Developing a numerical swallowing simulation

Tarun Sukumar, Ilias-Nikolaos Pergantis, Nathan Clarkson, Alex Hill
Dr A. Gambaruto
Department of Mechanical Engineering, University of Bristol

April 2019

ACKNOWLEDGEMENTS

The group wishes to thank Dr. A. Gambaruto for suggesting this investigation, assisting with the datasets plus for supporting the work throughout the length of this project. Additionally, our thanks goes to A. Bates for providing valuable industry insight and Dr. Y. Inamoto of the Fujita Health University in Japan for the CT images used throughout this project. We also wish to acknowledge my gratitude to the authors of the journals and books listed below.

DECLARATION

The accompanying Group Industrial Project report entitled: "Developing a numerical swallowing simulation" is submitted in the fourth year of study towards an application for the degree of Master of Engineering in Mechanical Engineering at the University of Bristol. The report is based upon independent work by the candidates. All contributions from others have been acknowledged above. The supervisor is identified at the start of the report. The views expressed within the report are those of the authors and not of the University of Bristol.

I hereby declare that the above statements are true:

.....
(Tarun Sukumar, Ilias-Nikolaos Pergantis, Nathan Clarkson, Alex Hill)

DECLARATION OF COPYRIGHT

Certification of ownership of the copyright in a dissertation presented as part of and in accordance with the requirements for the Final Degree of Master of Engineering at the University of Bristol, Faculty of Engineering.

I hereby assert that I own exclusive copyright in the item named below. I give permission to the University of Bristol Library to add this item to its stock and to make it available for consultation in the library, and for inter-library lending for use in another library. It may be copied in full or in part for any bone fide library or research worker on the understanding that users are made aware of their obligations under the copyright legislation, i.e. that no quotation and no information derived from it may be published without the authors prior consent.

.....
(Tarun Sukumar, Ilias-Nikolaos Pergantis, Nathan Clarkson, Alex Hill)

WORK ALLOCATION

Work Allocation		
Name	Project Contribution	Sections Written
Nathan Clarkson	Manually segmented all 22 images in ITKSNAP. Compared finite difference schemes to choose the best for automatic segmentation. Developed a method for automatically segmenting the bolus in 3D. Validated segmentation method by looking at the derivatives in 1D for a slice of the bolus. Assisted in the development of the moving boundaries for the MPS method and assisted in developing the morpher.	2
Tarun Sukumar	Created and developed a morpher to interpolate between the 22 segmented images of the bolus and the throat geometry to create a smoother moving boundary. Additionally, integrated the output to be compatibile with the both STARCCM+ and Matlab. Also improved the autonomous features and the flexibility of the code. Assisted with the integration of the csv files with the MPS solver and STARCCM+. Aided with the automatic segmentation as well	1,3
Illias-Nikoloas Pergantis	Developed a novel MPS Lagrangian solver. The solver was applied successfully on a number of benchmark cases. The solver was used for a static simulation of the bolus in the mouth using the segmented geometry. Using the morphed time steps a moving boundary method was created and improvements suggested. Aided with noise filtering in the automatic segmentation.	4,5,8.1
Alex Hill	Developed marco to automate mesh morphing process within STARCCM+. Undertook validation simulations to ensure usefullness of models that were used. Ran simulations using bolus geometry and produced a morphing model for the swallowing process. Helped create CSV files necessary for STARCCM+ morphing macro.	6,7,8

.....
(Tarun Sukumar, Ilias-Nikolaos Pergantis, Nathan Clarkson, Alex Hill)

Summary

A simulation of a healthy human swallow has been developed with the aim of improving the diagnosis and treatment of Oropharyngeal Dysphagia. Additionally, there is a strong emphasis on automating the diagnosis process. To facilitate this, an automatic method to segment the bolus using CT scans has been developed alongside a morpher which increases the temporal resolution of the dataset. Two numerical methods, STAR-CCM+ and Moving Particle Semi-Implicit (MPS), are employed to analyse the data and provide meaningful numerical data (i.e. pressure, velocity) to doctors. Important results, such as the volume fractions of the bolus, are derived which provides doctors with key information on the chance of aspiration; these data have been previously unavailable using current diagnosis methods.

List of Figures

List of Tables

Contents

1 INTRODUCTION

1.1 What is Dysphagia?

Oropharyngeal Dysphagia (OD) is a swallowing disorder which affects many processes including mastication, sucking, saliva management and control over the food path through the oral cavity, pharynx and oesophagus entrance. It additionally impairs the protection of airways and may lead to aspiration [1][2][3]. It affects individuals to varying degrees, chronic to transient, and can result from physical abnormalities, acquired, neurologic or physical dysfunction, progressive degeneration and psychiatric disorders [4]. OD is prevalent in patients with conditions such as stroke, Parkinson's disease, head and neck cancer and other neurodegenerative diseases [4][5]. This amounts to people with OD as a secondary condition to psychiatric problems are 43 times more likely die from choking [6].

The effects of OD are diverse; from mild discomfort to total inability to swallow and breathe. The consequences therefore include asphyxiation, malnutrition, and pneumonia. The most ubiquitous strategy in place to treat OD is bolus modification. This includes altering the pH, softening, and thickening to make food easier to swallow. Thickening aids swallowing by slowing the rate of movement of the bolus for an uncoordinated swallow. This increases the time for the larynx to be protected by the epiglottis and decreases the risk of aspiration [7].

1.2 What is a normal swallow?

Swallowing is a physiological function carried out over 500 times per day by an average human [8]. One swallow requires the cooperation of 30 muscles located within the oral cavity, larynx, pharynx, and oesophagus (See figure 2). Muscle movements are controlled by the peripheral nerves and several cranial [9]. In addition to physical deformities, any damage to these will result in problems with swallowing [10]. It is therefore important to understand the anatomy and neurophysiological control of a normal, healthy swallow.

Bones and Cartilage, Salivary gland and Teeth Bones, salivary gland and teeth are critical for mastication. Saliva, in particular aids, with bolus formation and transport. The salivary gland produces 95% of the saliva with the rest produced by glands in the lining of the oral mucosa. Whereas, the cartilages support the muscles and help in the pharyngeal bolus transport. The epiglottis is key in directing the bolus away from the airway and into the oesophagus.

Spaces and Muscular Control The upper aerodigestive tract is split up into 4 spaces: the oral cavity, oropharynx, nasopharynx and hypopharynx. There are other spaces which make no contact with the bolus like the laryngeal vestibule and laryngeal ventricle. The presence of bolus in these regions after a swallow indicates dysphagia [11]. All 30 muscles are striated apart from the medial and distal oesophagus which have parts that are completely smooth [12].

Stages The mechanism of swallowing can be split up int 4 stages: (1) oral preparatory stage, (2) oral transport stage, (3) pharyngeal phase and (4) oesophageal phase [9][13]. The 2 primary phases are voluntarily controlled whereas the latter 2 are involuntary. The timings and movements can vary in each stage, for example, the elderly take longer to trigger the first 2 voluntary stages than the youth [13]. This report only considers the bolus after the oral transport phase.

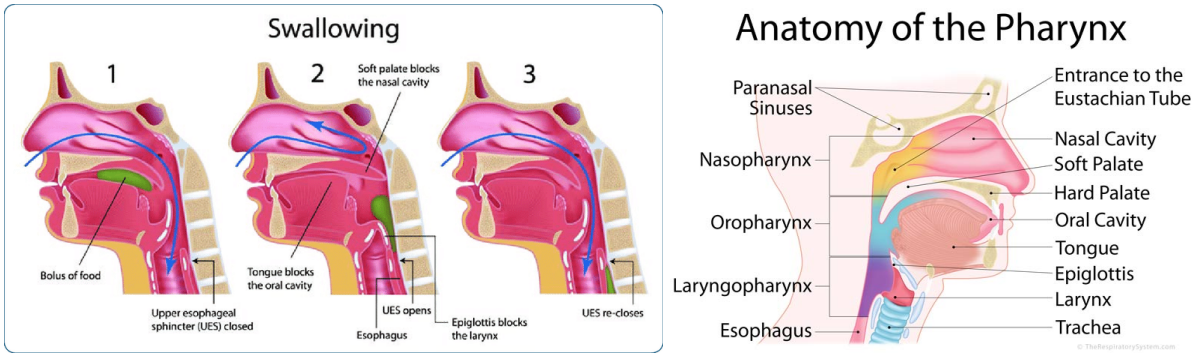


Figure 1: Stages of Swallowing

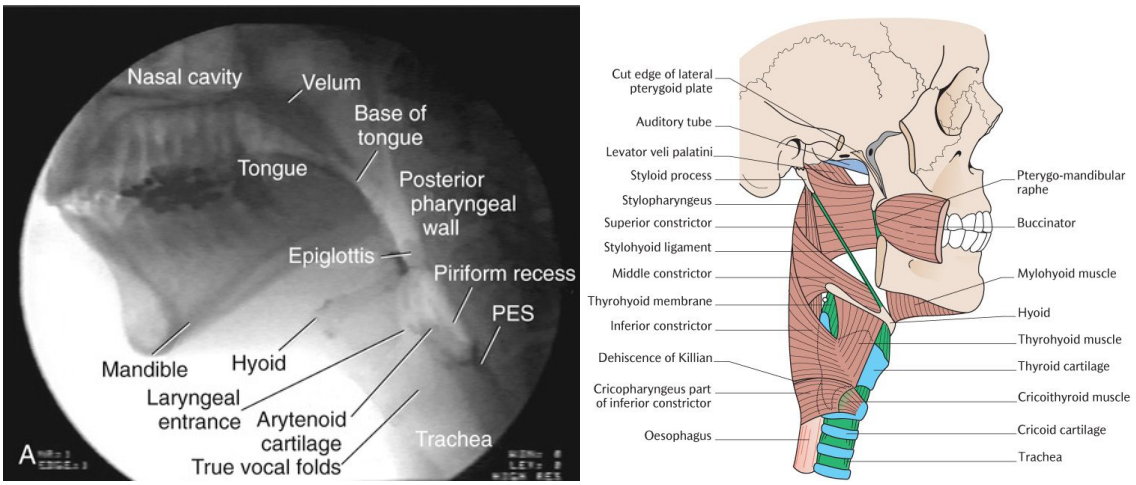


Figure 2: Pharyngeal muscles

1.2.1 Oral Prepartory Phase

The primary objective of this stage is to carry out mastication and breakdown food to form a cohesive bolus [14] [13]. Once mastication is complete, with the help of the saliva, and a bolus is formed, the bolus is stored between the tongue and the hard palate.

1.2.2 Oral Transport Phase

The bolus is now forced into the oral cavity at the back and then to the oropharynx. The soft palate is elevated and seals of the nasal cavity from the oropharynx [14]. These oral nasopharyngeal seals are important for generating closed pressure loops in the oral cavity, pharynx, and oesophagus. During the oral transport and pharyngeal phases, the sequential points of high pressure resulting from the pharyngeal movements, aid the transport of the bolus through the aerodigestive tract.

The contraction of the superior longitudinal muscle holds the bolus in between the lingual dorsum and hard palate. This raises the anterior tip and lateral edges of the tongue toward the alveolar ridge. When these are in contact, the tongue prepares to initiate the posterior transport of the bolus. The tongue presses against the hard palate and propels the bolus into the oral cavity by moving in a wavelike form. The combination of intrinsic and extrinsic muscles of the tongue direct the bolus into the oropharynx. The posterior tongue depresses, and the soft palate elevates allowing a smooth passage for the bolus into the oropharynx [15].

1.2.3 Pharyngeal Phase

After the oral transport phase, the bolus is transported into the oropharynx and passes the anterior faucial pillars; this contact triggers the involuntary phase [9] (See figure 1 and 2). It is not a true reflex since there is a certain amount of control on how the bolus is transported by the previous phase, known as the leading complex. Properties such as texture, taste and volume all modulate the timing of the action [16].

The pharyngeal phase stops respiration to protect the airways by contracting the laryngeal musculature. The pharynx is then raised by contracting the pharyngeal muscles while the tongue is retracted to towards the rear pharyngeal wall [17]. The pharyngeal constrictors are activated in a rostral-caudal direction and descends from the nasopharynx to the upper oesophageal sphincter (UOS) in a wave-like fashion, the pharyngeal peristalsis [18]. This wave strips along the tail of the bolus and squeezes it through the pharynx to the upper oesophagus.

The pharynx elevates while the suprahyoid muscles contract moving the hyoid bone superiorly and anteriorly. Simultaneously, the thyrohyoid muscles contracts to move the larynx towards the hyoid bone. These movements are crucial to the swallowing process; they provide protection for the airways by moving the larynx under the tongue base and inverting the epiglottis to direct the bolus away from the laryngeal inlet. Also, the laryngeal elevation and the elevation of the hypopharynx creates a negative pressure point below the bolus which induces a sucking effect to help with the transport of the bolus [18]. Thirdly, the elevation of the larynx and pharynx creates a force which pries the cricoid cartilages away from the pharyngeal wall and pulls open the cricopharyngeal muscles and the UOS. This adds to the negative pressure gradient to aid the transport of the bolus. This, in addition to the driving force from the tongue and pharyngeal stripping wave, means that the bolus transport achieves great efficiency [18].

1.2.4 Oesophageal Phase

The bolus passes through the UOS and the relaxation begins which lasts between 0.5s to 1.2s. The bolus now has entered the oesophagus and the cricopharyngeal muscles return to its contracted state. This seals off the oesophagus and prevents the bolus moving back into the hypopharynx. The oesophageal peristalsis is activated which forces the bolus down into the lower oesophagus sphincter and the stomach. The oesophageal peristaltic wave squeezes the bolus down to the oesophagus and a secondary wave traverses this to reinforce the effect [19].

1.3 Current Diagnosis Techniques

Videoflouroscopy (VF) and Videoendoscopy (VE) are the most widely used diagnosis techniques for analysis in a morphological and kinematic manner. Magnetic resonance imaging (MRI) and computed tomography (CT) are common for assessing pathological conditions.

During a videoflouroscopy swallow study (VFSS), the patient swallows a fluorescent substance, barium sulfate, and its high contrast ratio allows for 2D visualisation of swallowing. This is used on adults and children as the standard diagnosis as it can provide an in-depth view on all 4 stages of swallowing as well as, the accurate detection of aspiration. The exposure to ionising radiation is tackled by employing techniques such as tight coning and pulsed fluoroscopy.

During videoendoscopy, a fiberoptic tube is inserted through the nasal cavity. The dynamics of the swallow are difficult to assess with this procedure because of the tumultuous upheaval of

the larynx and pharynx temporarily obscures any view. Any aspiration that occurs during the swallow can usually be identified immediately after by the residue remaining in the airway.

Conventional CT (Computed Tomography), which produces detailed cross-sectional images and 3D images, does not have sufficient temporal resolution to show very quick swallowing movements and therefore has not been utilised in swallowing research. However, this has recently changed with the newest scanner, MSCT-320. This device is equipped with 320 rows of 0.5mm detectors along it's body axis and can acquire a volume data set covering a maximum range of 16cm. This range can cover the area of the oral cavity, pharynx, larynx, and upper esophagus. The CT images used in this report were provided by Y. Inamoto of the Fujita Health University. There are 22 frames of 360°, with a time step of 0.175s, of a 35 year-old healthy volunteer, swallowing 10 ml of honey-thick liquid (5 % v. barium). The size of each frame is $512 \times 512 \times 320$ voxels and the voxels are $0.468 \times 0.468 \times 0.5mm$.

1.4 Motivation

Using videofluoroscopy, the only data that are gained are the images themselves. Also, there are currently no other literature which captures how the bolus moves numerically. By incorporating CFD, it is possible to get more quantitative data which means the medical problem can be better understood and treatments can be improved and tailored to individual patients. Finally, the project also hopes to reduce the amount of time between initial consultation and treatment by automating as many of the steps in the process as possible to reduce the impact of dysphagia.

1.5 Project Aims

The first part of the project is to develop a way to automatically segment the bolus in the images. A method of morphing between each segmentation, so that there are more images with a smoother transition, will then be established. Two computational fluid dynamics (CFD) solvers will then be used to simulate the bolus movement. The 3D morphs will be used to create meshes that can be used in STARCCM+. A novel Lagrangian Moving Particle Semi-Implicit solver will be developed and benchmark tests will be carried out. This solver will then be used to simulate the swallow in 2D.

2 SEGMENTATION

In order to analyse the movement of the bolus during the swallowing process, a geometrical representation of the bolus must be formed. This is done by segmenting the bolus part of the image, thus creating a 3D geometry for each of the 22 CT scans; these images capture the different stages of the swallow. These 3D geometries are then ready to be morphed to create a smoother transition between each image. An automatic segmentation method is developed in this section so that the bolus mass is able to be morphed faster.

Manual segmentation can easily be done using a program such as ITK-SNAP [20]. This is very time consuming and not very accurate because there is no clear boundary for some parts of the bolus. Manual segmentation is also made more difficult due to the fact that the CT images contain a lot of noise and the bolus is only visible over a small number of voxels. This process can be improved by using a semi-automatic tool but this only uses the intensity value in the image for segmentation and therefore it is still inaccurate.

2.1 Automatic Segmentation Method

An edge-based method is used to automatically segment the bolus. This method uses the idea that there is a large change in the intensity of the image at the bolus boundary. At the point on the boundary, the second order derivative of the intensity will be equal to zero because there is an inflection point. The method used is shown in a paper by Gambaruto [21].

$$\tilde{\mathbf{n}}^{\mathbf{T}}\mathbf{H}\tilde{\mathbf{n}} = 0 \quad \text{where,} \quad \tilde{\mathbf{n}} = \frac{\nabla I}{|\nabla I|} = \begin{pmatrix} \tilde{n_x} \\ \tilde{n_y} \end{pmatrix} \tag{1}$$

Where $\tilde{\mathbf{n}}$ is the unit normal vector and $\mathbf{H}$ is the hessian.

The edges of the bolus are found by locating the inflections (where equation 1 is equal to zero). But firstly, a method to calculate the first and second order derivatives needs to be established. This is explored in subsequent sections.

2.2 Finite Difference Schemes

Several Finite Difference (FD) schemes are compared to determine which is best suited for calculating the derivatives of the intensity. The FD scheme must be able to calculate these accurately while suppressing high frequency noise, primarily due to the high amount of noise present in these images. The images also have a low spatial resolution which means the FD scheme must be able to determine the derivatives accurately over a small number of points.

2.2.1 Spectral Analysis

The FD schemes that are investigated are all anti-symmetric filters of type III as can be seen in equation 2 [22].

For a general FD scheme:

$$\frac{\sum_{k=1}^{M} c_k(f_k - f_{-k})}{h} \text{ where, } M = \frac{N-1}{2} \tag{2}$$

Where N is the window length, h is the point spacing, c_k is the coefficient for each point, and k is the point in the finite difference scheme.

The frequency response is [23]:

$$H(\omega) = 2i\sum_{k=1}^{M} c_k sin(k\omega) \tag{3}$$

Where $H(\omega)$ is the frequency response and ω is the frequency. To observe the behaviour of an FD scheme equation 3 can be used to plot to the frequency response.

2.2.2 Central differences

To determine the weightings of each node in the central differences approximation (for a simple 3 point stencil), a Taylor series expansion is taken about point U_i .

$$U^{i-1} = U^i - \Delta x U_x^i + \frac{1}{2}\Delta x^2 U_{xx}^i - \frac{1}{6}\Delta x^3 U_{xxx}^i + O(x^4) \tag{4}$$

$$U_i = U_i \tag{5}$$

$$U_{i+1} = U^i + \Delta x U_x^i + \frac{1}{2}\Delta x^2 U_{xx}^i + \frac{1}{6}\Delta x^3 U_{xxx}^i + O(x^4) \tag{6}$$

Where U^{i-1} is the value of the function at a point $-1\Delta x$ in the x-direction , U_x^i is the derivative with respect to x of U^i, Δx is the spacing between points and $0(x^4)$ is the 4th order error.

The finite difference approximation has the form:

$$U_x^i \approx \frac{1}{\Delta x}\sum_{k\in\ell} c_k U_k \tag{7}$$

Where ℓ is the stencil, which is a list of points that are used in the finite difference approximation, k is the point in the stencil and c_k is the weighting coefficient for each point.

For each Taylor expansion, the coefficient of the derivatives can be combined to find the weightings.

$$U_x^i = (c_{i-1} + c_i + c_{i+1})\frac{1}{\Delta x}U^i + (-c_{i-1} + c_{i+1})U_x^i + (\frac{1}{2}c_{i-1} + \frac{1}{2}c_{i+1})\Delta x U_{xx}^i + O(x^3) \tag{8}$$

This leads to a set of simultaneous equations that can be solved to find the weighting for each node.

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} c_{i-1} \\ c_i \\ c_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{9}$$

2.2.3 Smooth noise robust differentiator

The Smooth Noise Robust Differentiator (SNRD) is designed to be used for applications with noise [24]. Instead of determining the weightings using a Taylor series expansion, they are calculated from a desired frequency response.

If a filter is defined as length N (odd), filter coefficients c_k and is a function of values sampled at N points, which are a distance h apart, it can be stated that:

$$f_k = f(x_k),\ x_k = x^* + kh,\ ,k = -M,...,M \text{ and } M = \frac{N-1}{2} \tag{10}$$

Where k is the point and x^* is the inquiry point. A general form of the numerical derivative can then be defined as:

$$f'(x) \approx \frac{1}{h}\sum_{k=1}^{M} c_k.(f_k - f_{-k}) \tag{11}$$

Where f_k is the value of the function at position k.

The SNRD aims to be precise on low frequencies and to have guaranteed suppression of high frequencies. This is done by setting the frequency response to be equal to that of the ideal differentiator for $\omega = 0$ and to set the frequency response to zero for the highest frequency, $\omega = \pi$. Therefore, a set of simultaneous equations are obtained, with definitions holding from equation 3:

$$\begin{cases} \left.\frac{\partial^i H(\omega)}{\partial \omega^i}\right|_0 = \frac{\partial^i H_d(\omega)}{\partial \omega^i} & i = 0,1,2,...,n \\ \left.\frac{\partial^j H(\omega)}{\partial \omega^j}\right|_\pi = 0 & j = 0,1,2,...,m \end{cases} \tag{12}$$

The frequency response of the system is the same as in equation 3. Because it is a sinusoidal function, it is zero for either $\omega = 0$ or $\omega = \pi$ for i=0. Therefore, if $N = 5$, $n = 3$ and $m = 2$, there are only two equations that are non trivial:

$$2icos(0)\sum_{k=1}^{2} kc_k = i \quad 2i\sum_{k=1}^{2} kc_k cos(k\pi) = 0 \tag{13}$$

Solving these equations gives $C_1 = \frac{1}{4}$ and $C_2 = \frac{1}{8}$. Both n and m can be varied to determine the order of accuracy of the FD scheme and the number of nodes. The 2nd order derivatives can also be derived if it is assumed that the frequency response of a second order system is:

$$H(\omega) = c_0 + 2i\sum_{k=1}^{M} c_k cos(k\omega) \tag{14}$$

2.2.4 Comparison of FD schemes

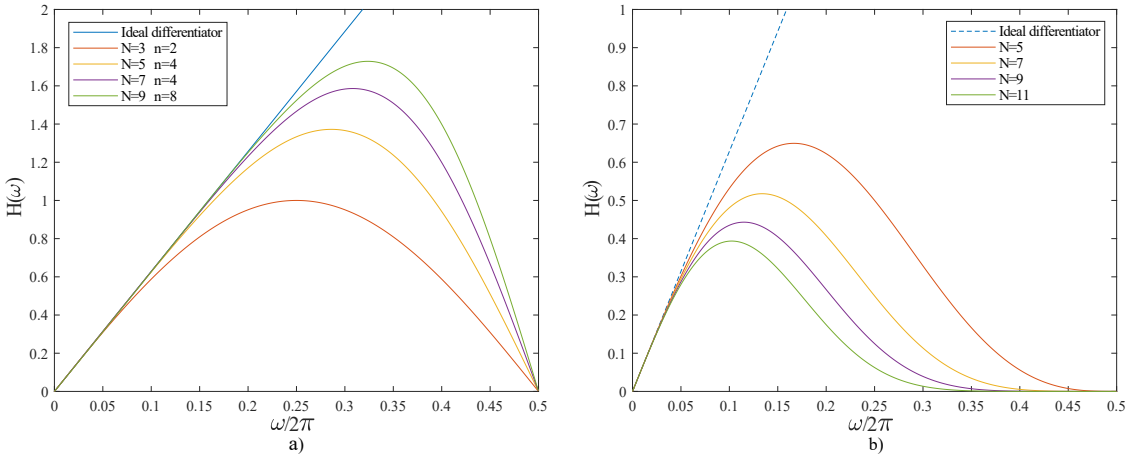


Figure 3: The frequency response of a) the central differences finite difference scheme and b) the smooth noise robust differentiator (n=2) where N is the number of points in the stencil and n is the order of accuracy

It can clearly be seen from figure 3 that for the same window length, the central differences curve approximates the ideal differentiator more accurately that the SNRD. On the other hand, the SNRD is evidently better at suppressing high frequency noise, which can be seen by the fact that $H(\omega)$ is zero for high values of ω in figure 3b. Whereas, $H(\omega)$ is only zero when $\omega = \pi$ for the central differences approximation in figure 3a.

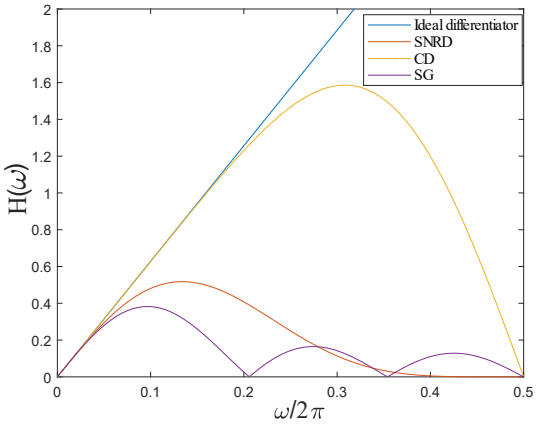


Figure 4: The frequency response of the SNRD (2nd order accurate), CD (6th order accurate) and Savistky Golay (2nd order accurate) filters for a 7 point stencil

Figure 4 compares a different numerical method, know as the Savitsky Golay (SG) filter, for calculating numerical derivatives[25]. It can be seen from the graph that this method is as accurate as the SNRD and also has some form of filtering to remove noise. Unlike the SNRD, the SG filter does not guarantee noise suppression for all high frequencies because the frequency response shows a sinusoidal response. This means that with the presence of noise, as in the medical images, the derivatives will be calculated more accurately using the SNRD.

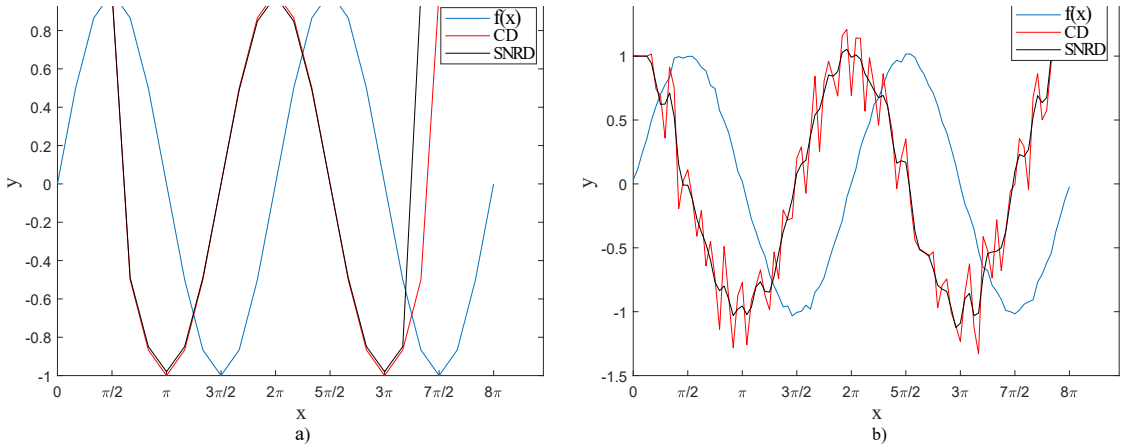


Figure 5: The first order derivatives of $sin(x)$ calculated using the central differences approximation (N=9, n=8) and the SNRD (N=9, n=2) a) without and b) with noise where the spacing between points is a) $\frac{\pi}{12}$ and b) $\frac{\pi}{48}$. The noise was created by adding a random number between 0.02 and -0.02 to the sine wave

It can be seen from figure 5a that without the presence of noise, the CD approximation is marginally closer to the exact value of the derivative than the SNRD. On the other hand, figure 5b shows that the SNRD copes with added random noise much better than the CD approximation. This is important as even though the function doesn't look like it has significant amount of noise, it still has a large affect on the derivatives. Noise creates additional inflection points which means it would be significantly harder to detect an edge. Therefore, for the purpose of edge detection, the SNRD is used.

It is also important to note that the spacing between the points is large in figure 5a. This is important as the bolus only appears over a small number of voxels. With a large spacing, the SNRD is close to the exact derivative so can be used for the automatic segmentation.

2.3 Implementation

The 12th image is used as a test for automatic segmentation because it is the hardest to manually segment. This is due to the fact that the intensity values inside the bolus are not significantly higher than the surrounding voxels. This frame illustrates the transition between the pharyngeal and the oesophageal phase - the most complex part of the swallow.

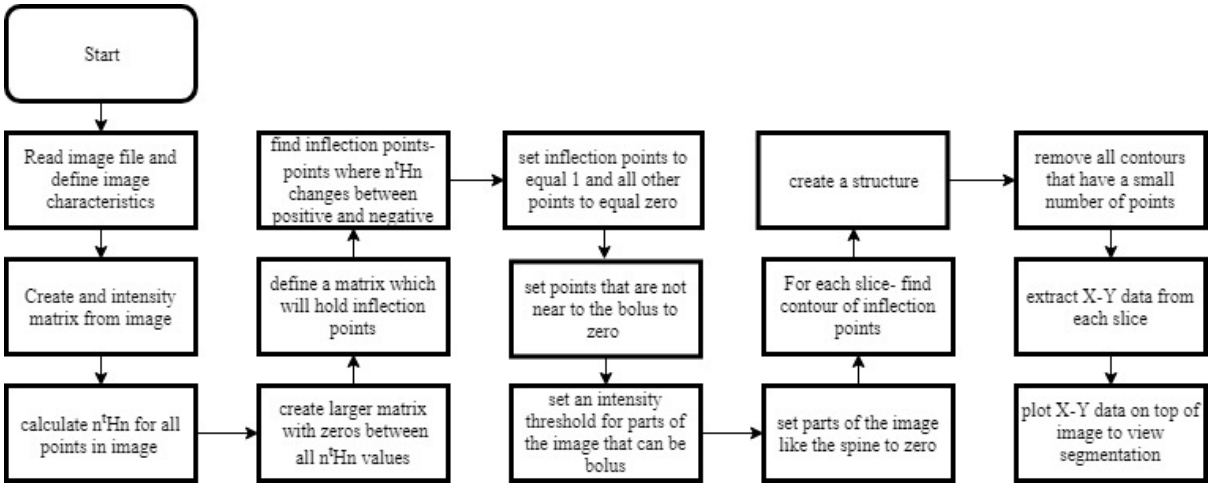


Figure 6: A flow chart to show the method used to segment the images

Figure 6 shows the pseudocode used to create the automatic segmentation. It is important to note that step 4 re-sizes $\tilde{n}^T H \tilde{n}$. This is illustrated in equation 15. This is done so that if an inflection point is between two points in the image, it can be placed in one of the added points in the matrix.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{15}$$

Noise is removed for each 2D slice in steps 12 and 13. This is done by creating a structure from the contours and then removing all of the contours below a threshold amount of points.

2.4 Results

Figure 7 shows the automatic segmentation of the bolus in the middle plane of the 12th image. As the order of accuracy of the SNRD increases, there is a significant increase in the accuracy of the segmentation. When the SNRD is only accurate to 2nd order, there are significant sections of the bolus that are missing from the segmentation. The difference between the 4th and 6th order segmentations is smaller than that between the 2nd and 4th order. This implies that 6th order is of suitable accuracy for the segmentation.

It can also be seen from figure 7 that as the window length increases the amount of bolus in the segmentation increases. The segmentations at higher order are also significantly smoother. This is because as the window size increases, because more noise is filtered out, there are fewer misplaced inflection points.

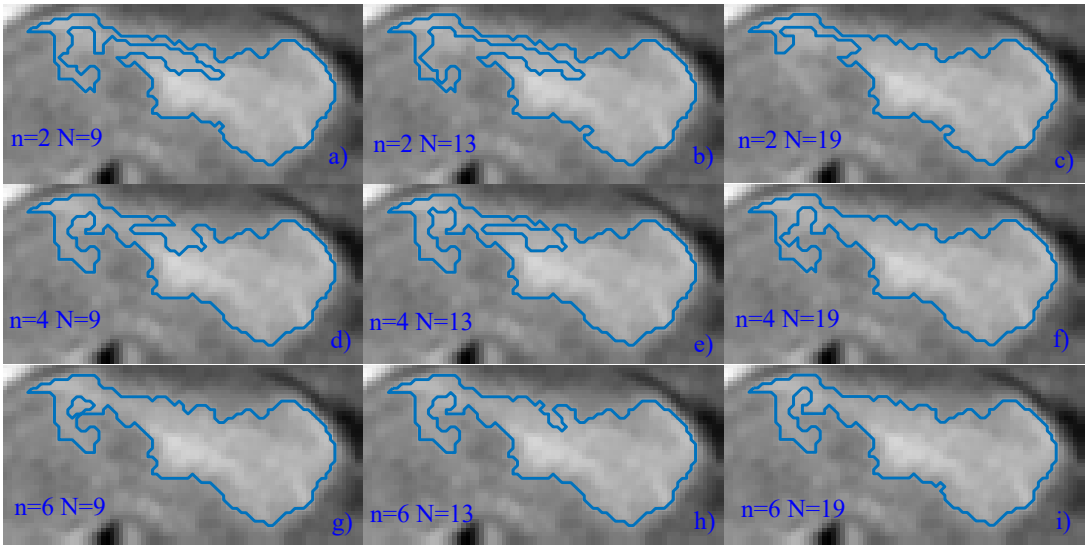


Figure 7: The segmentation of the bolus in image 12 with different SNRD where n is the order of accuracy of the filter and N is the window length of the filter. The 2nd order derivatives are 1 order of accuracy higher than the first order derivatives

The segmentation in figure 7i is the best because it has both the highest order of accuracy and largest window length. This has resulted in it containing the most amount and bolus and being the smoothest segmentation.

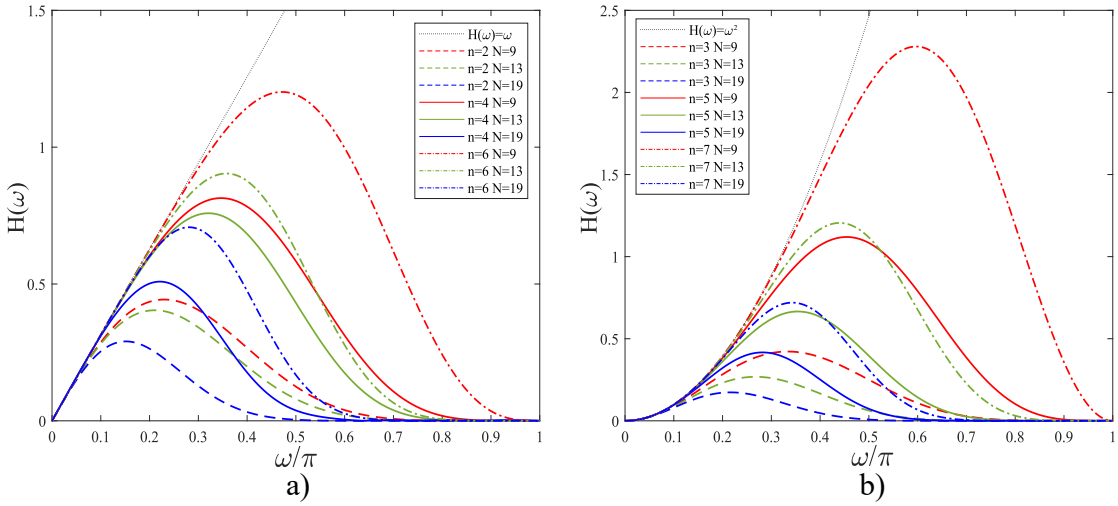
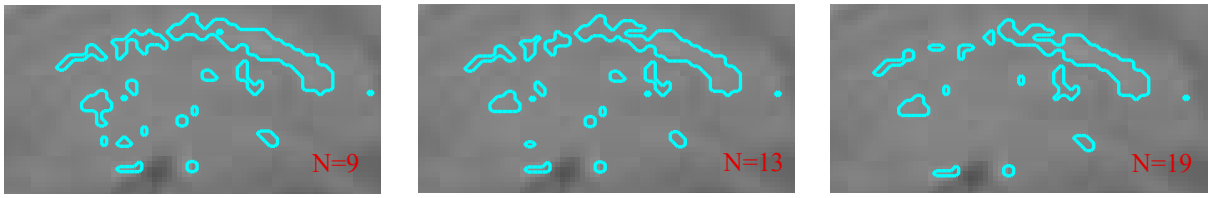


Figure 8: The frequency response of the a) first order and b) second order SNRDs where n is the order of accuracy and N is the window length of the filter

Figure 8 shows the frequency responses of all of the SNRDs that are used to generate the results in figure 7. Some of the frequency responses have similar shapes to each other. For example, figures 7e and 7f have similar frequency responses and therefore yield similar segmentations. It is clear from figure 8 that the best finite difference scheme for a noisy image is one that has a frequency response that converges to zero at a frequency as low as possible. However, this is only true to a limit because if a low enough frequency is suppressed, then it is possible that important details from the image can be lost.

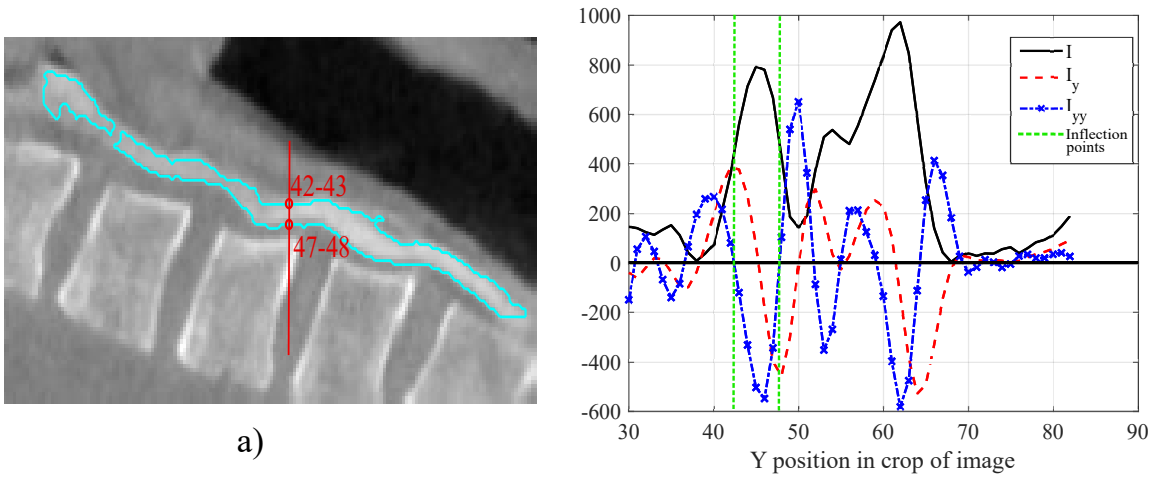


a) b) c)

Figure 9: The automatic segmentation of the bolus in a plane that is close to the edge of the bolus for image 12 using an SNRD where n=6 and a) N=9 b) N=13 c) N=19

The evidence in figure 9c shows that the ideal filter for one slice of an image does not necessarily work for all slices of the image. For a window length of 19 the image is smoothed significantly. With this amount of smoothing the bolus has been blurred into the background because the original image does not have a clear visual boundary. Consequently, it could be useful to be able to change the SNRD depending on the location in 3D. This would take some manual input, but would mean the results are more accurate. Figures 9a and 9b have considerably smaller window lengths and show much more clear edges to the bolus. Also, unlike the central slices, more of the bolus is captured for a smaller window length.

The method for removing noise would still not work for these slices as the bolus is not just one area. It would not be possible to remove the noise by removing small contours in the structure as some parts of the bolus would be removed. The noise is not removed in figure 9 to show that the bolus contours are similar sizes to the noise contours. The number of slices where this is a problem is small; the overall volume of the bolus is therefore not affected significantly by the inaccuracies at the edge. Hence, a small amount of manual segmentation would still be required to remove noise at the edges of the 3D volume.



a) b)

Figure 10: The a) segmentation of the bolus of image 15 in the middle slice and b) graph of intensity (I) for y positions along the line marked in a) and the first and second order differentials with respect to y. The inflection points at the edges of the bolus are marked

Figure 10 validates the mathematical justification of the segmentation. The points marked on figure 10a are the the inflection points on the graph in figure 10b. These points correspond to the edges of the bolus. The second order differential crosses the x-axis between the two points which means the inflection point is between these in the image. It is therefore important to increase the resolution of the image to facilitate this. Figure 10b) also shows that there

are many inflection points on the image which are noise. Some of these inflection points are below the threshold intensity for them to be boundaries of the bolus so this noise is removed automatically.

3 IMAGE MORPHING

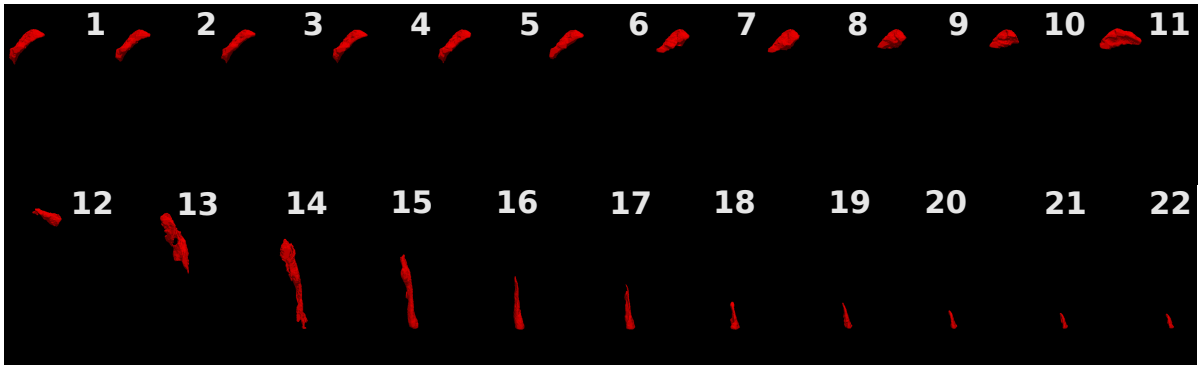


Figure 11: STL segmentations of the bolus mass; the timesteps to interpolate between

The current data from the CT scans provides 22 frames over the span of the swallowing stages. This dataset has very poor temporal resolution and thus is not fit for use in Computational Fluid Dynamics (CFD) analysis since the timesteps are too big. This section focuses on interpolating between each of these 22 frames to accurately determine the boundary profile of the bolus at new timesteps; this will make it more manageable to perform CFD calculations. By finding out how the bolus morphs between the frames, it can be determined how the bolus moves and calculate the interpolations by applying a fraction of the morph. Therefore, the program will be able to control the new timestep according to the output required.

Figure 11 shows that the movement of the bolus is subtle for the first 10 frames. However, from observing the step between frames 12 to 13, the necessity to derive intermediate frames is seen. To achieve stability in CFD calculations, the timesteps need to be at least of the order 10^{-4} whereas, the raw data is of the order 10^{-2}. This section details how the morph between frames is found and the subsequent preparation of data for CFD. Furthermore, the section explores the level of automation the morpher achieves and its usefulness to investigate OD. Figure 12 additionally shows the position of the bolus relative to the throat boundary in the key phases of swallowing.

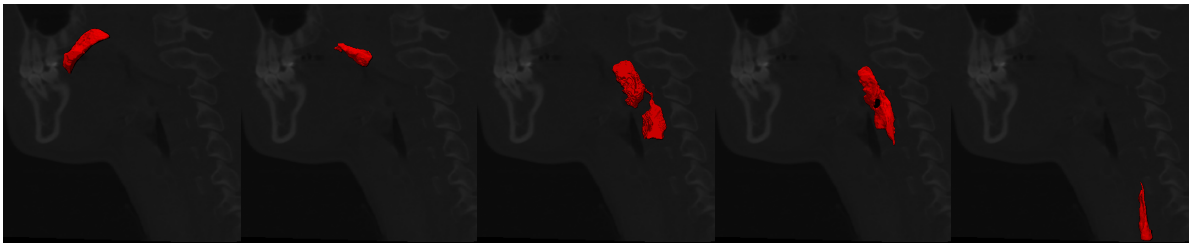


Figure 12: Position of the bolus at key phases of the swallow

3.0.1 Bicubic convolution algorithm

The morpher employs the Bicubic convolution algorithm to interpolate between the frames. The section below details the mathematics behind calculating the coordinates of the bolus boundary in new timesteps.

If the function values f and the derivatives f_x, f_y and f_z are known at four corners of the unit square, the interpolated surface can be written as:

$$p(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \tag{16}$$

The bicubic spline interpolation requires the result of the above linear system for all grid cells [26]. An interpolator, with the aid of a convolution with this kernel in each dimension, can be defined:

$$W(x) = \begin{cases} (a+2)|x|^3-(a+3)|x|^2+1 & \text{for } |x|\leq 1, \\ a|x|^3-5a|x|^2+8a|x|-4a & \text{for } 1<|x|<2 \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where a is set as -0.5 or -0.75. Keys proposed that $a=-0.5$ produced a third-order convergence with respect to the sampling interval of the original function and therefore it is more commonly used [26]. Considering the case when $a=-0.5$, the piecewise function is more useful, by substituting a, rewritten as:

$$p(t) = \frac{1}{2}\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}0 & 2 & 0 & 0\\ -1 & 0 & 1 & 0\\ 2 & -5 & 4 & 1\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}f_{-1}\\ f_0\\ f_1\\ f_2\end{bmatrix} \tag{18}$$

for a value of t between 0 and 1 for one dimension. 4 sampling points are required for the calculation in each dimension (2 to the left and 2 to the right). t is denoted as the distance between point 0 and the inquiry point [26].

3.1 Method

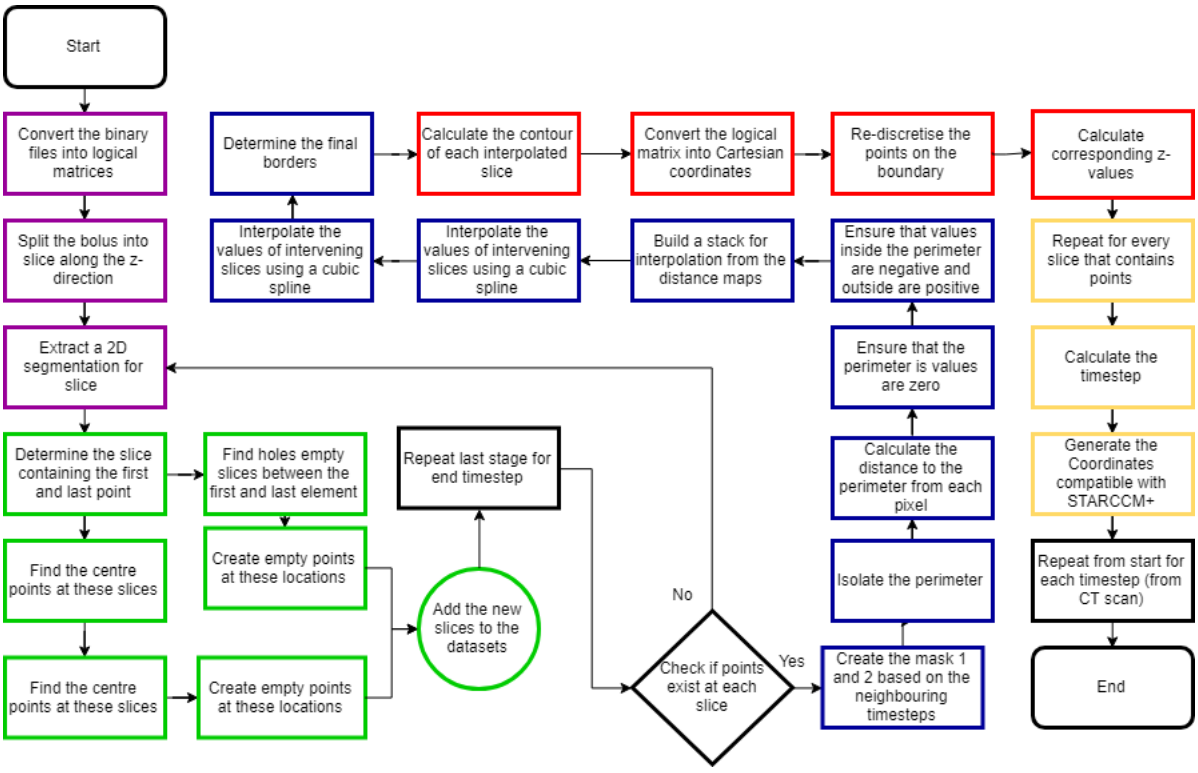


Figure 13: Pseudo-code presenting the steps taken by the morpher

The morpher splits the bolus at each timestep to 221 2D slices along the z-axis. The amount of slices the data is split into is controlled by the resolution of the raw CT scan data - the morpher tries to maintain this spatial resolution through all processes the data is subjected to. By interpolating between the timesteps for each slice, the movement of the 2D boundary is obtained. The 2D slice can then be concatenated to form the boundary of the 3D bolus. The program takes advantage of Matlab's compatibility with binary files to manipulate binary datatsets. This is detailed further in the pseudocode, figure 14. Each colour in the pseudo-code signifies the operations of a single Matlab function. Other features of the code are to smooth the results as well as repair the boundaries where the data has been discretised in an unsuitable manner. The program at the end yields a ".csv" file with the boundary points of the bolus for each timestep in an XYZ format. This is both compatible with STARCCM+, Meshlab and Matlab. A modified version of the program repeats the process for morphing of the throat boundary instead.

The morpher has been designed to run automatically after the segmentation stage. It remains flexible as parameters such as the number of points on the boundary and the number of interpolations can easily be adjusted.

3.2 Validation

The morphing tool that has been developed has to be tested to see if the results generated are accurate and valid. This test aims to validate the process by considering a known morphing test case and comparing it to the results ascertained from the program. If the results are the same or the differences between them are explainable, the program can be deemed as valid. The test case under question is the morphing from a diamond to a square. The program is set to interpolate 10 frames between mask 1, the diamond, and mask 2, the square.

The literature details how the morphing path between a diamond and a square can be obtained by expressing them as circles but with different powers as shown in equation 19 [27]. The circle can be expressed using the Euclidean distance from its centre ($n = 2$) whereas the diamond can be expressed using the Manhattan distance (based on the first power instead, $n = 1$) [28]. Thus, the 'near-perfect' square can be expressed using the fourth power, $n = 4$. Plotting the 'circles' based on the first power to the fourth power (in small fractional steps) will yield the morph for a diamond to a 'near-perfect' square.

$$x^n + y^n = const. \tag{19}$$

Where n is the fractional power.

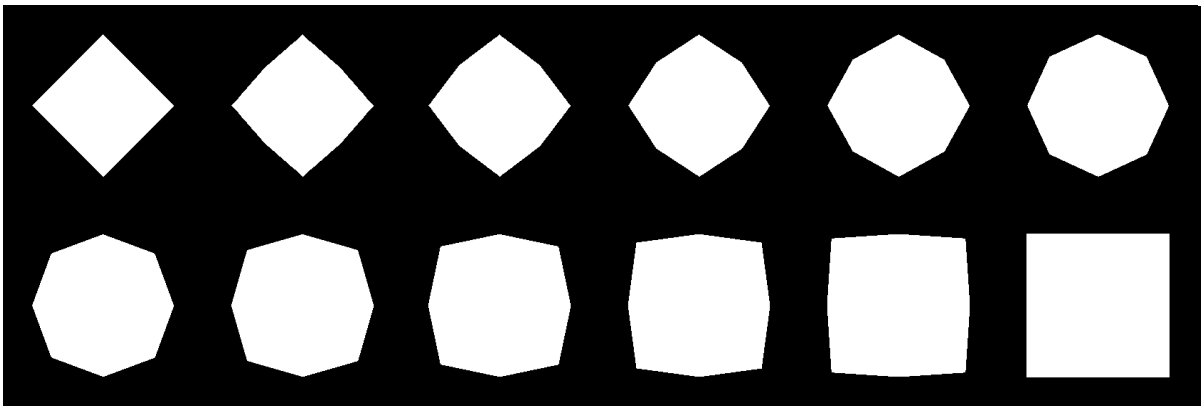


Figure 14: 2D Morphing results from a diamond to a square

The validation yields the successful transformation from a diamond to a square, as per literature, as the middle frame approximates to a circle. If the number of interpolations is to be increased, the middle frame will tend to a perfect circle which matches up with literature. The points on the square move in a logical manner between the rest of the frames and thus it can be concluded that the morphing program is working in an appropriate fashion.

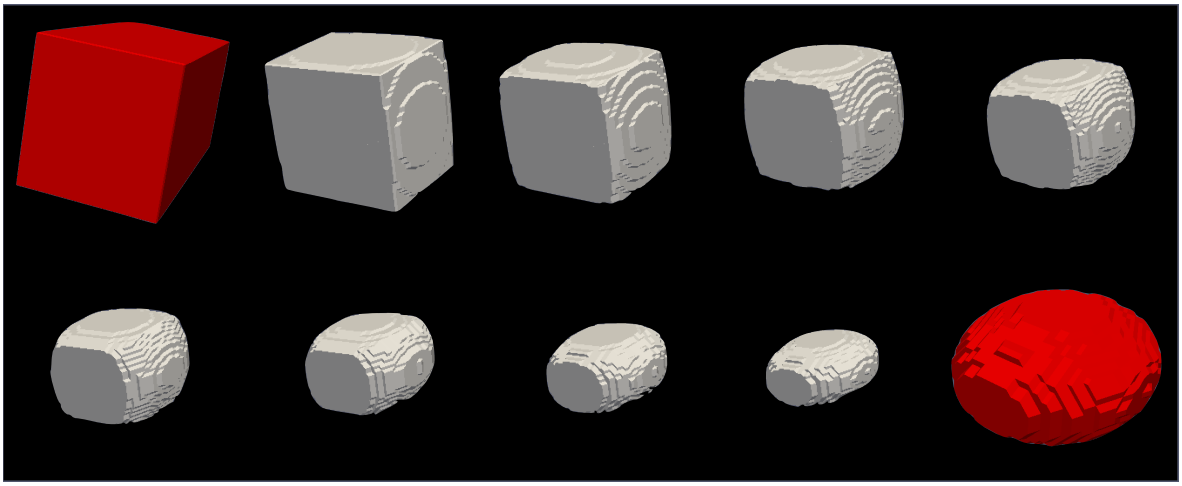


Figure 15: 3D morphing test from a large cube to a smaller ellipse

It is important to validate cases in 3D; especially between objects with varying with width, height and/or length. This introduces the idea of morphing between objects with differing number of slices in the z-direction to see how the morpher copes with this. This test will yield valuable information on how the morpher overcomes problems with mass conservation.

Figure 15 illustrates the morphing between a large cube and a smaller ellipse. The morphing achieved between the two objects is smooth and consistent with the literature which demonstrates that the morpher is able to handle 3D objects with differing dimensions. Therefore, it is ready to be applied to the swallowing simulations.

3.3 Results

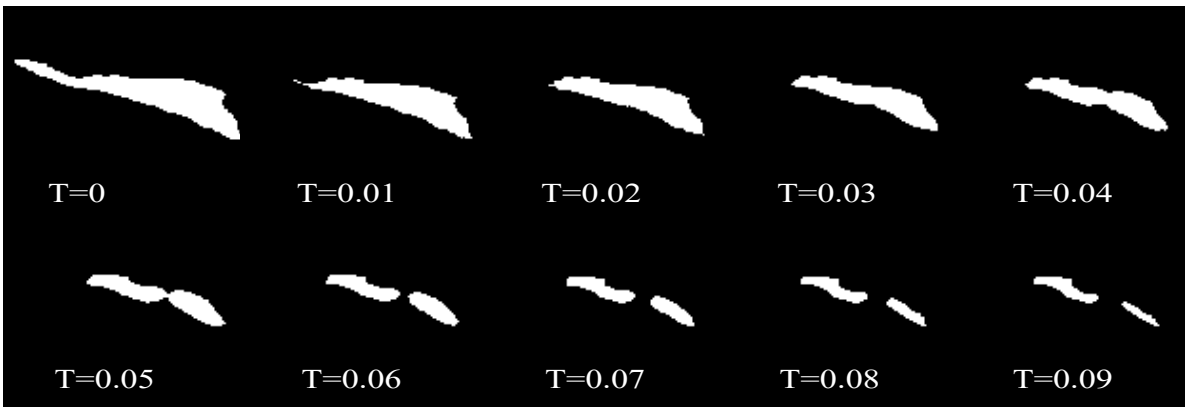


Figure 16: 2D morph results of a section of the bolus

The results in figure 16 depict the transformation from mask 1 at $T = 0$ to mask 2 at $T = 0.09$. This was a 2D section from the middle of the bolus between timesteps 16 and 17 used as an example to illustrate the capability of the morpher. The equivalent throat geometry was also tested to see how the morpher deals with a boundary as opposed to a object, figure 17.

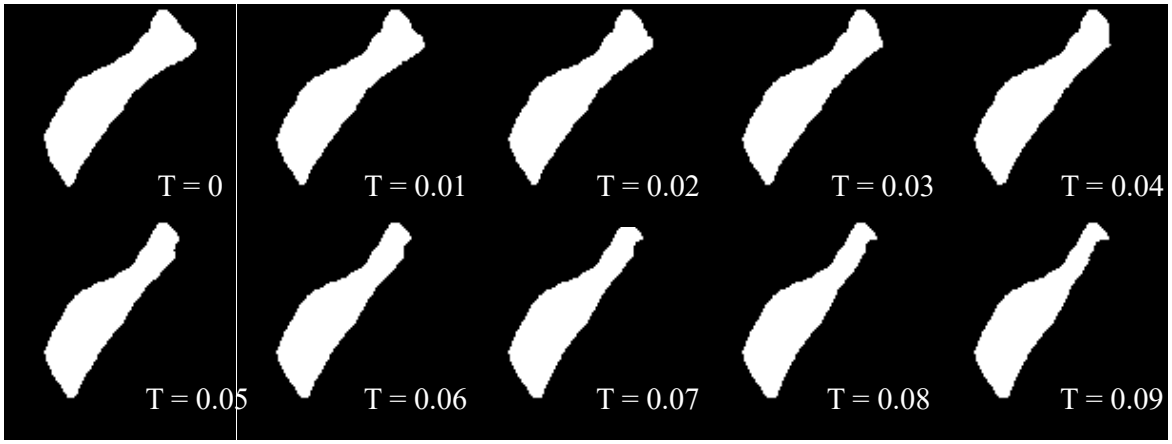


Figure 17: 2D morph results of a section of the throat boundary

Since the swallowing procedure is split up into stages, the bolus motion will drastically vary. It is therefore important to observe the motion through a transition in phases as well as the bolus movement in a single phase only. For the figures below showcasing the 3D interpolations of the bolus mass, the objects in blue show the initial and final frames from the CT scans and the red objects show the interpolations calculated by the morpher.

Figure 18 shows the bolus moving down the oesophagus after the pharyngeal phase. The only movement happening is the redistribution of bolus mass as it flows down the oesophagus; there is minimal bolus movement observed in the bolus in a single swallowing phase.

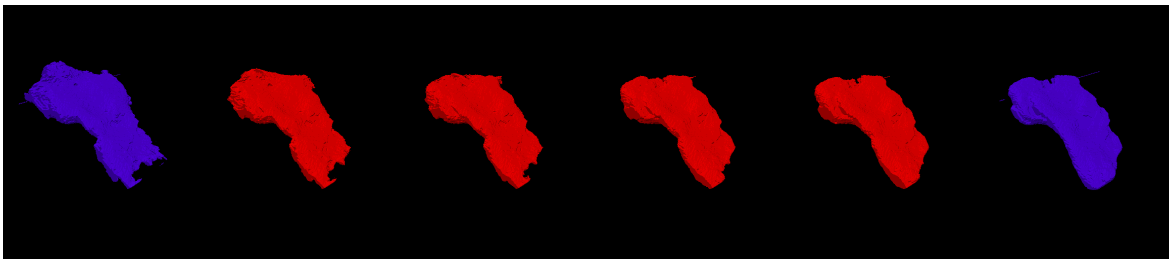


Figure 18: Interpolation of results between frames 14 and 15

In contrast, figure 19 shows the key transition between the pharyngeal and oesophageal phase. It can be observed how the morpher tracks the bolus through this phase transition.

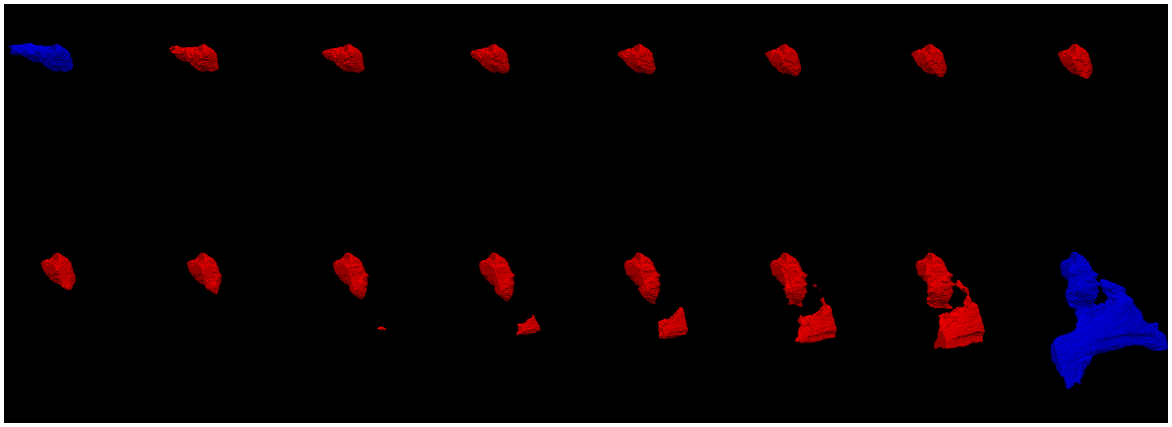


Figure 19: Interpolation results between frames 12 and 13

The final part of the pharyngeal stage is where the elevation of the hypopharynx creates a negative pressure point below, which sucks the bolus down away from the laryngeal inlet. This

is illustrated in first 9 frames of figure 19 as the interpolations shows the bolus subtly moving along the throat boundary. Next, there is a rapid change in bolus geometry where the throat squeezes the bolus down the oesophagus. The next 5 frames depict the actions of the muscles relaxing and oesophageal peristaltic wave transforming a compact, almost spherical bolus to a thin film as it forces the bolus down the oesophagus [18].

3.4 Discussion

The interpolations obtained between the frames 14 and 15 are smooth and the transformation of geometry corresponds with how the bolus is expected to react to the forces applied by the peristaltic wave. The interpolations are observed to be evenly spaced with respect to time. The minimum cell quality acceptable is 10^{-8} for valid CFD results. The mesh created from the results has minimal stretching and the faces that are created are even. Therefore, the mesh passes the threshold comfortably and thus a good mesh has been generated, see figure 34.

The interpolations between frames 12 and 13 also behave and move concurrently with forces the epiglottis and the hypopharynx are applying. Additionally, the geometry created produces an excellent mesh. However, it is important to note some inaccuracies with the transfer of bolus mass from a compact, spherical like structure to a thin film. The way the morpher interpolates between each timestep means that the thin film originates unconnected to the bolus implying that mass is being created. The bridge between the 2 masses is so thin that it is not significant enough to appear. The bolus splitting up, as it does between frames 12 and 13, presents the hardest challenge with respect to interpolations and as proven, the morpher has been able to produce physically meaningful results, apart from the aforementioned issue. The quality of the mesh obtained from the results is good and therefore, valid results from the CFD can be expected. Interpolations between frames 12 and 13 specifically highlight the importance of bridging the gap between the data obtained from the CT scans. The bolus splitting up is not captured at all in the initial 22 frames.

Future Work

One way to address the issue surrounding the splitting of the bolus is to obtain a dataset with a greater spatial resolution. The current morpher would be able to interpolate more accurately and data, such as the bridge between the two bolus masses, will not be lost. But obtaining data of a higher resolution is not a feasible solution to the problem. Resizing the image to artificially enhance the image resolution is possible way this can be resolved. However, this will decrease the accuracy of the bolus boundary. The most efficient way of tackling this problem is by, instead, changing how the morpher captures the motion from one frame to the next. Current literature employs a free-from deformation (FFD) to achieve this instead.

3.4.1 Free-form Deformation

Parametrisation of complex objects has been an important area of research and of great interest to many fields; parametrisation of wings and aerofoils for shape optimisation in potential flows and design of cardiovascular devices in viscous flows are a few examples. FFD has expanded from its original use with solid modelling systems to encompass such problems. While the FFD's counterparts directly distort and deform the object, FFD manipulates the lattice built around it which allows it to impact the whole space the object is embedded into. The lattice

topologically relates to a cube in 3D and rectangle in 2D, illustrated by figure 20. FFD can handle surface of any formulations and it is independent to the mesh applied to it [29].

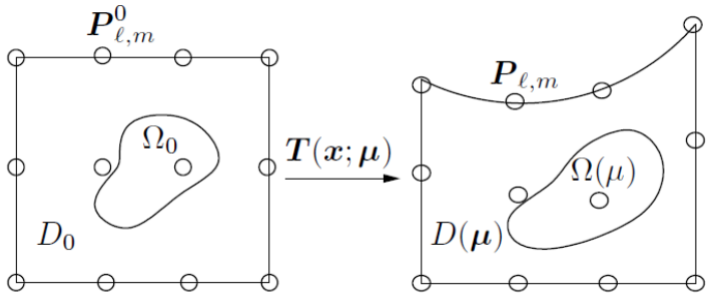


Figure 20: Impact of the global transformation matrix from the FFD

A valuable attribute of this method is how the computational grid is being deformed when the space around (or inside) the object is being deformed; this is particularly useful for automated design optimisation algorithms.

FFD can be applied locally or globally without compromising the shape smoothness. FFD is made up of a time-consuming offline stage as well as an online stage. Therefore, the offline stage can be calculated and stored whereas, the shorter online part can be repeatedly calculated and used to update the whole sum [29].

Once the global transformation matrix is known, the interpolations can be obtained by applying percentage magnitude of the transformation matrix to the original unperturbed matrix. This allows interpolations to be obtained and hence, is a very powerful method. Allowing the mesh to deform means the action of the epiglottis on the bolus can be better replicated and it will allow for even more accurate interpolations.

4 DEVELOPMENT OF MPS FLOW SOLVER

4.1 Motivation behind Lagrangian Solvers

The motion of the bolus in the throat is governed by the incompressible Navier-Stokes equations. These equations can be solved in Eulerian or Lagrangian reference frame. In Eulerian frame the domain is discretised in a number of stationary regions where the flow properties are computed, usually employed by commercial CFD solvers. When the flow geometry is rapidly changing (i.e. in the case of the intricate movements of the pharynx walls) Eulerian approaches require re-meshing which is very limiting. A Lagrangian reference frame treats the flow as a number of individual fluid particles. A fluid property at a particular particle is computed based on the surrounding particles. The advantage of this is that no mesh is required. Lagrangian methods are an ongoing field of research, with applications in modelling Tsunami waves and avalanches, topics previously inaccessible to CFD techniques [30]. In the context of this report two fluid simulation techniques were developed, MPS Solver and STARCCM+.

4.2 Outline of MPS Method

The outline of the following MPS scheme was first introduced by Koshizuka and Oka in 1996 [31]. In Lagrangian techniques, the fluid is discretised in a number of fluid 'lumps'. The mass of each fluid particle is given by the total fluid mass divided by the number of particles. Thus,

if their number is maintained during a simulation, mass is automatically conserved. Considering mass and momentum conservation, the Navier-Stokes equations in a Lagrangian reference frame can be written as:

$$\frac{Du}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2 u + g \tag{20}$$

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla \cdot u = 0 \tag{21}$$

where $\frac{D}{D_t}$ denotes the material derivative, ρ the density, P the pressure, g the gravitational acceleration and u the fluid's velocity field. Assuming a continuous velocity field, the velocity of a particle at the next time instance can be written as $u^{n+1} = u^* + u^{**}$. The superscript $n+1$ indicates the velocity field after an instance Δt and $*$,$**$ the predictor and corrector velocity respectively. The location of the particles at the next iteration is computed using $x^{n+1} = x^n + \Delta t u^{n+1}$ [32]. By substitution of the velocity expression into equation 20 along with using forward time differencing, equation 22 can be derived:

$$u^* + (u^{**} - u^n) = -\frac{\Delta t}{\rho}\nabla P + \Delta t(\nu\nabla^2 u + g) \tag{22}$$

Finally, by using the Helmholtz Decomposition Theorem the two velocities are found as:

$$u^* = u^n + \Delta t(\nu\nabla^2 u + g) \tag{23}$$

$$u^{**} = -\frac{\Delta t}{\rho}\nabla P \tag{24}$$

Pseudo Compressibility for Explicit Pressure Calculation

Having derived equations 23 and 24, a coupling to the pressure-gradient field is required. Firstly, the density field is re-written as $\rho = \rho^* + \rho^{**}$. Then, direct substitution of this expression into equations 21 and 24 leads to the following Poisson equation:

$$\nabla^2 P_i = -\frac{\rho}{\Delta^2 t}\frac{n_i - n_0}{n_0} \tag{25}$$

where $n \propto \rho$, is a discrete equivalent of the density for the fluid particles, each denoted by i. This implicit equation has the form of $Ax = b$ which in most conventional schemes is solved by methods such as the Conjugate Gradient, the Gaussian Elimination or the SOR. A disadvantage of these methods is that the computation time of solving the Poisson equation increases with $N^{\frac{3}{2}}$, where N is the total number of flow particles [33]. Due to this, an explicit scheme was developed in Oochi 2010 based on a pseudo-compressibility formulation of the flow [34]. Therefore, the pressure can be re-written as:

$$P_i = c^2\rho\frac{n_i - n_0}{n_0} \tag{26}$$

where c is the speed of sound. This formulation of the flow allows for a computational time scaling of the order N. For this reason, this method has been implemented in the computer game industry for real time fluid movement. The correct selection of the speed of sound is of utmost importance for a realistic pressure field. There are conflicting views in the literature on the choice of c, some authors suggest $c = const$, while others use a Mach number definition such that, $c = \frac{1}{Mach}u^n_{max}$ or u^*; both choices are investigated.

Weight Function

Particles interact with each other inside a circular region. The radius is denoted by r_e, known as the radius of compact support. This is scaled based on the particle spacing at the start of the simulation, l_0. A higher value of r_e leads to a more over-determined system of equations and thus, aids the robustness of the simulation and smooths the results. Although, it also increases the computational time. For all simulations, r_e was set to $3l_0$ and $l_0 = 4$mm. A weight function, w, is used in Lagrangian techniques to vary the impact of one particle to another based on their distance. The weight function adopted for this solver is shown in equation 27:

$$w(r, r_e) = \begin{cases} 0 & r > r_e \\ \frac{r}{r_e} + \frac{r_e}{r} - 2 & r \leq r_e \end{cases} \tag{27}$$

where r is the distance between two particles. This particular form was selected over the usual choice of $\frac{r_e}{r} - 1$, due to it having a continuous derivative with respect to position when $r = r_e$ [35]. A singularity exists in w at $r = 0$; this aids inter-particle repelling when the distance is small.

MPS Variables and Particle Types

The discrete particle density n for a particle i is expressed as:

$$n_i = \sum_{j \neq i}^{N} w(|r_{ij}|, r_e) \tag{28}$$

n_0 from equation 26, is computed at the start of the simulation as the initial packing density of the fluid particles. This is the decompressed state where pressure is zero. In order to reduce the computational costs, a neighbor list (surrounding particles) is created for each particles which is updated whenever the particles move.

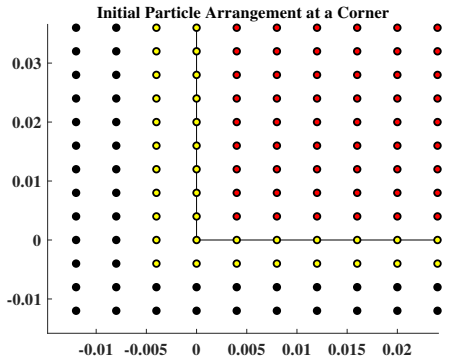


Figure 21: Particle distribution at a Cartesian corner, the solid line indicates the boundary

Three particle types were used in the scheme: fluid, wall and dummy particles which are denoted by red, yellow and black respectively in the scatter plots. For the wall particles a Dirichlet velocity condition is used. In the Dam Break case, the velocity is zero whereas for the swallowing simulation it can be set to the speed of the morphed boundary. Pressure is computed at wall particles but the gradient pressure field is not.'Dummy' particles were used to pad the region around the wall. A 'no penetration' boundary condition is achieved because, as fluid particles approach the wall particles, n increases and they are repelled. The particle domain near a boundary is shown in figure 21. Two layers of wall particles were used [33].

Differencing Schemes

The differential operators in the MPS method are computed as a weighted sum of the surrounding particle's properties. The gradient operator of a property, ϕ, for a particle, i, is given by:

$$\langle \nabla \phi \rangle_i = \frac{d}{n_i} \sum_{j \neq i}^{N} w(r) \frac{(\phi_j - \phi_i)(x_i - x_j)}{|x_i - x_j|^2} \tag{29}$$

where, d is the number of dimensions [31, 33]. The operator for the Laplacian is given by:

$$\langle \nabla \phi \rangle_i^2 = \frac{2d}{\lambda n_i} \sum_{j \neq i}^{N} w(r)(\phi_j - \phi_i) \tag{30}$$

where λ is a function used to convert from a continuous to a discrete solution[34]. The two operators used in the conventional explicit method are equation 29, for the pressure, and equation 30, for the Laplacian of the velocity. Recent papers have suggested that a more stable approach is to solve equation 22 implicitly [32]. Substituting the expression for the Laplacian to the RHS of equation 22, the predictor velocity linear system can be written in the following form:

$$u_i^* = u_i^n + \Delta t g + \frac{\nu 2d\Delta t}{\lambda n_i^n} \sum_{j \neq i}^{N} w(r)(u_j^* - u_i^*) \tag{31}$$

By applying a Dirichlet boundary conditions at the wall, the final system is:

$$u_i^* \left(\frac{\lambda n_i^n}{\nu 2d\Delta t} + n_i^n\right) - \sum_{j \neq i}^{N_f} (w(r) u_j^*) = \frac{\lambda n_i^n}{\nu 2d\Delta t}(n_i^n + \Delta t g) + \sum_{i \neq j}^{N_w} (w(r) u_j^*) \tag{32}$$

where the superscript n denotes the current time instance; f and w stand for fluid and wall particles and j for the neighbours of particle i. This system at each time step is solved using the biconjugate gradient stabilised method at a tolerance of 1E-8. With respect to the pressure gradient computation, Oochi10 uses equation 29 to reduce computational costs [34]. However, it was found that the resulting pressure field of that method had a high level of noise and is prone to instability. To overcome this, a least squares scheme was implemented to compute the pressure gradient. Such a scheme has uses a 2D Taylor expansion around any particle for a given region defined by r_e. In matrix notation, this system is written as $S\delta f = f$, where, $f = f_i - f_0$, δf is a vector of derivatives and S is a matrix of Taylor coefficients. The equation can be normalised by dividing by the radius of compact support to increase the smoothness of the solution. The normalised equation is written as:

$$W \bar{S} L^{-1} \delta f = W f \tag{33}$$

where $\bar{S}$ is the normalised coefficient matrix, L is a diagonal length normalisation matrix and W is a diagonal weight function matrix. The literature recommends a weight function of the form $(\frac{1}{r})^d$ and it was found that $d = 4$ gives the maximum error reduction [36]. The inverse of $W\bar{S}$ was computed using a Singular Value Decomposition (SVD) algorithm.

Least Squares Results

The developed Least Squares (LS) code was applied to a complex sinusoidal function. The 2D gradient field is shown in figure 22. The error in the dominant 1st order derivative direction was

0.64% for a 100x100 grid and 2.4% for a 50x50 grid. These values agree with the computed errors of other authors [36].

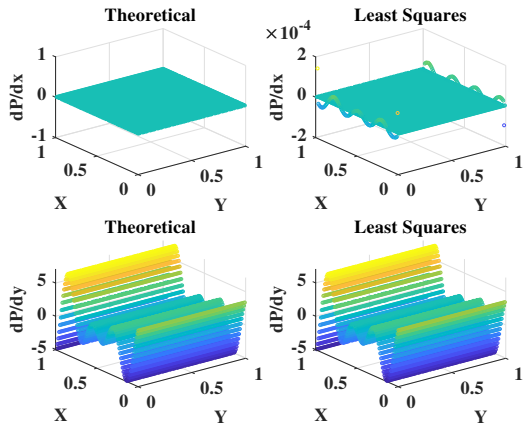


Figure 22: Comparison of the LS results with the theoretical derivatives

Although the LS scheme performed well against a number of test functions, stability issues occurred when implementing it to the solver. A portion of the project time was spent in understanding the instability sources. It was identified that they were caused by f (pressure difference), taking both positive and negative values. This introduces both repulsive and attractive forces between particles, deteriorating the stability of the solver. To cope with this phenomenon, the following pressure gradient formulation was adopted:

$$\langle \nabla P \rangle_i = \frac{d}{n_i} \sum_{j \neq i}^{N} w(r) \frac{\Big((P_j + P_i) - (min(P_i) + min(P_j)) \Big)(x_i - x_j)}{|x_i - x_j|^2} \tag{34}$$

where, $min(P_i)$ signifies the smallest pressure in the neighborhood of particle i [32, 37].

5 BENCHMARK CASES

5.1 Stationary Tank

5.1.1 Stability Criteria

The first investigation is a static tank simulation. Since the current proposed scheme is novel in literature in combing an implicit diffusion (expected to increase stability) to an Artificial Compressibility (AC) explicit pressure scheme, a stability scheme had to be developed. Initially, a customary CFL scheme is used for the selection of dt in the form of:

$$\Delta t = \alpha min(\Delta t_{convection}, \Delta t_{diffusion}) \tag{35}$$

where $\Delta t_{convection}$ and $\Delta t_{diffusion}$ are the convection and the diffusion time steps and α is a safety factor ≈ 0.1. In most studies, $\Delta t_{diffusion}$ is defined as either $\frac{max(\lambda)}{2d\nu}$ or $\frac{\tau_{diff} l_0^2}{2\nu}$, as before, d stands for the number of dimensions, ν is the kinematic viscosity and τ_{diff} is a scaling factor ≈ 0.4. For water at room temperature, $\nu \sim 10^{-6} m^2 s^{-1}$. For 2D and the aforementioned grid, this leads to $\Delta t_{diffusion} = 7.4 sec$ for the first expression and $\Delta t_{diffusion} = 0.8 sec$ for the second ($\tau_{diff} = 0.1$). These are both extremely large values and thus are not useful. Other physical

flows such as in haemodynamics, have $\Delta t_{diffusion} = 1 \cdot 10^{-6} sec$ due to a smaller and more diffusive length scale [32]. For convection, the following stability criterion is used:

$$\Delta t_{convection} = \frac{|r_{min,rel}|}{|U_{max,rel}|} \tag{36}$$

where *rel* stands for the relative property values [34]. A relative frame between any two particles was selected because particles that get unnaturally close to each other create a spike in pressure. This usually occurs at the predictor step, when particles are moved to position r^* based on a u^* prediction. Having a CFL condition check based on equations 35 and 36 at the start of the iteration loop does not capture this. To deal with this, a Courant number for the flow is defined as:

$$Courant_{flow} = \frac{\Delta t_{loop} |U^*_{max,rel}|}{|r_{min,rel}|} \tag{37}$$

and the stability criterion can be expressed as:

$$Courant_{flow} < Courant_{limit} \tag{38}$$

Equation 37 is computed after any predictor calculation. If it is found that criterion 38 is not met, the step is recomputed using $\Delta t = \frac{\Delta t}{2}$. This is done iteratively until a stable Δt is found.

5.1.2 Constant Mach Number

The width of the water column was $0.1m$ and the height $0.2m$. The fluid particles were placed in a Cartesian grid with initial inter particle distance $l_0 = 4$ mm. The aim was to achieve the lowest total kinetic energy. Using $M = const.$, the speed of sound is scaled based on the speed of the flow. This is a more natural approach than using a constant speed of sound since, it allows scaling of the pressure field based on the velocity field. The simulation was run for Mach $0.2, 0.1, 0.05$. The particle velocity is non zero because it is an oscillating system; each particle is experiencing a gravitational force pulling it downwards and a balancing pressure force trying to keep it in position. The $Courant_{limit}$ was formulated as $\frac{U_{allowable} \Delta t_{start}}{l_0}$, with $U_{allowable} = 0.1 ms^{-1}$ and $\Delta t_{start} = 5 \cdot 10^{-5} sec$. For all Mach numbers initially, the column collapses as pressure is built at the bottom wall and spreads through the fluid. However, this configuration proves to be unstable because the particle velocity is too high before pressure is distributed. Even with unnaturally high velocities no wall penetration occurred.

5.1.3 Constant Speed of Sound

A constant speed of sound is used for immediate balancing of the gravitational force. The speeds of sound used are $c = 50, 100, 150 ms^{-1}$. The mean particle velocity for a 0.4x0.2x0.004m (2D) tank after 0.8s, is found to be $0.0446, 0.0402, 0.0446$ respectively. To compare the static performance of the simulation, a metric, which is the total kinetic energy of the system per unit volume, was introduced. This can be expressed as:

$$\bar{J} = \frac{J_{total}}{V_{total}} = \frac{\frac{1}{2} m_i \sum_i^{N_f} |u_i|^2}{V_{total}} \tag{39}$$

Table 1: Kinetic Energy per unit volume for various authors

Study	J (Joules)	$\bar{J}$ (Joules m^{-3})	$\nu(m^2s^{-1})$
Current ($c_{sound} = 100ms^{-1}, t = 0.8s$)	$2.58 \cdot 10^{-4}$	0.81	10^{-6}
Yamada ($t = 5s$) [38]	0.05	3.18	10^{-4}
Oochi ($t = 5s$) [38, 34]	10	637	10^{-4}

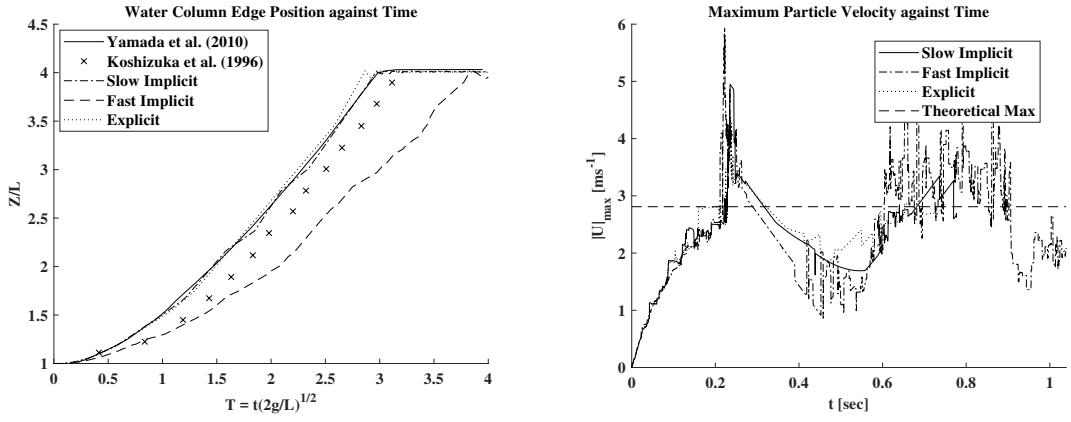
Table 1 shows the KE results for a number of studies. It can be seen that the energy per unit volume of the novel scheme outperforms all existing studies, even with a lower viscosity. This is attributed to the implicit diffusion formulation currently employed, stabilising the conventional fully explicit (both diffusion and pressure) schemes.

5.2 Dam Break

The next test is the collapse of a water column. The column has a height of 0.2m and a width of 0.1m. At $t = 0$ the wall that keeps the fluid in position disappears and the column collapses under gravity. The main metrics of the test are the leading edge position of the column and the maximum velocity developed, theoretically given by, $U_{max} = 2(gh)^{\frac{1}{2}} = 2.80ms^{-1}$ [39]. The particles were placed in a Cartesian grid, with interparticle distance $l_0 = 4$ mm and a total of 1250 fluid and 2056 wall and 'dummy' particles. A number of speeds of sound were used, however, it was found that maintaining Mach less than 0.05 leads to more stable simulations. As such, $c_{sound} = 100ms^{-1}$ was used. Three tests were run using these parameters, the first two looked at the current proposed scheme, where u^* is computed implicitly as before and using varying CFL conditions. The third computed the predictor step explicitly using equation 30. The conditions for the 3 tests are shown in Table 2.

Table 2: CFL parameters for the Water Collapse Tests

	Slow Implicit	Fast Implicit	Explicit
$\Delta t_{start}(10^{-5}s)$	4	6	4
CFL_check	0.0150	0.0250	0.0150
r_{min}	$l_0/10$	$l_0/10$	$l_0/10$
Δt (t>0.1), $(10^{-5}s)$	1	2	1
Δt (t>0.4), $(10^{-6}s)$	8	10	8



(a) (b)

Figure 23: A comparison of the water collapse metrics for various tests [38],[31]

The value of r_{min} was selected such that no two particles would be allowed to get too close to each other, which would destabilise the pressure field. At the end of corrector step, the time step is reset to either the initial value ($t < 0.1$) or to some arbitrary value in order to avoid

simulation bottle necking. The edge position results are shown in figure 23 along with the maximum velocity results.

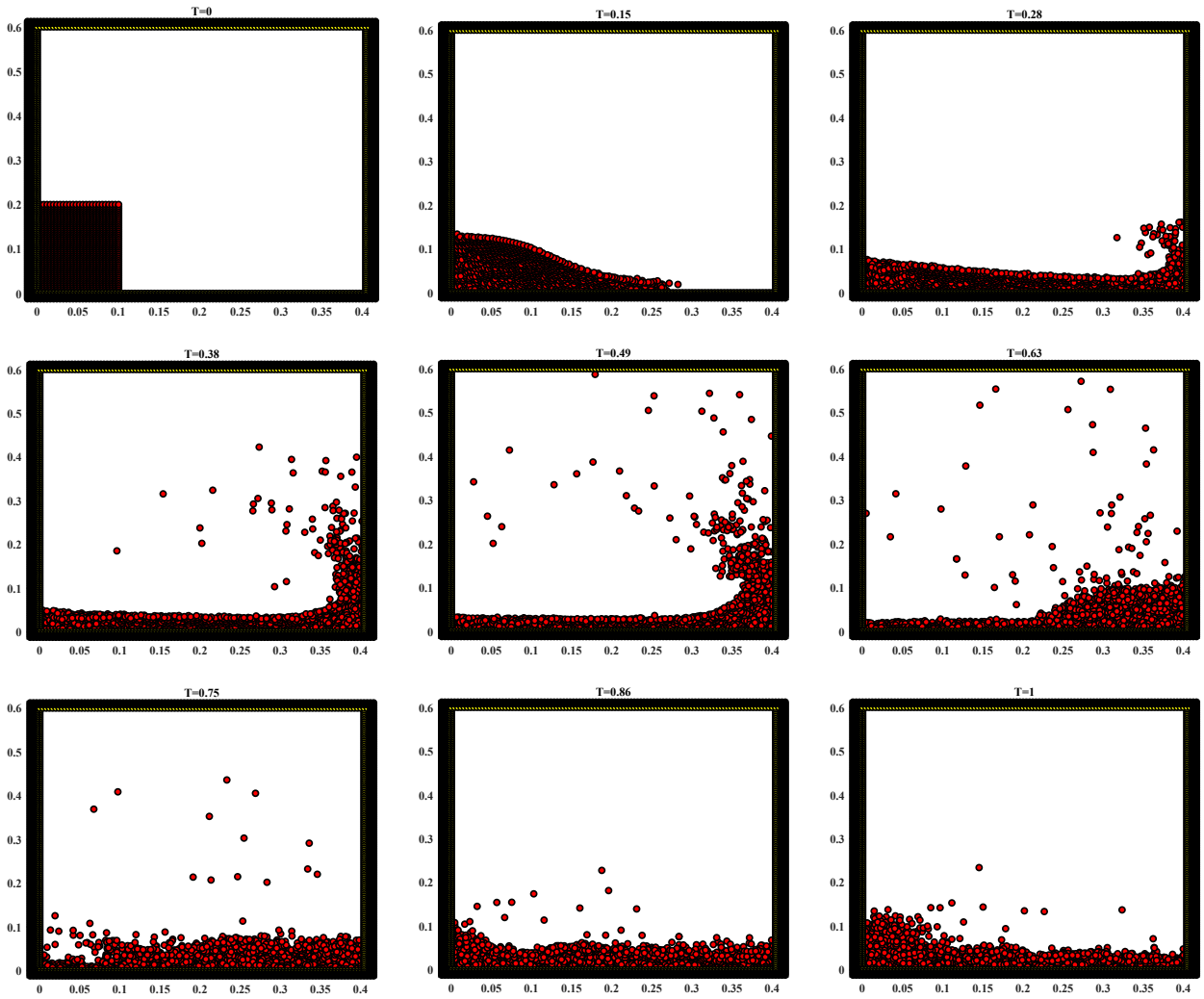


Figure 24: Various time instances of the Water Collapse test

In figure 23a it is shown that the leading position of the more strict implicit and explicit schemes perform closely to the literature. The less strict implicit scheme performs more closely to the experimental measurements than the others for $T < 1.5$. However, as time progresses it lags behind. This is attributed to the noise in the velocity field shown in figure 23b. The other two methods have smoother maximum velocity trends, which damp out the collision spike quickly. The water collapse for the less strict implicit scheme is shown in figure 24. An issue identified with the implicit solver is that equation 32 became unstable for particles which have less than three neighbors. These particles are eliminated from the linear system and treated as solid particles under gravity.

5.3 Creating a Complex Boundary

The boundaries used for the swallowing simulations were the segmented and morphed throat geometries of frames 6 and 7 of the medical images. Frame 6 is used to initialise the particles. An important metric of the MPS method is the initial particle spacing l_0. This is used to calibrate via n_0, the inter-particle pressure, as well as the pressure from the boundaries. Thus, in order to have a uniform repulsive force from a wall, the particles in the wall must be space

uniformly. The area of the fluid region is given by $A_{fluid} = N_{fluid} l_0^2$. Thus, defining η as the bolus fill portion of the throat area and A_{throat} as the segmented throat region, the particle spacing can be found:

$$l_0 = \sqrt{\frac{A_{bolus}\eta}{N_{fluid}}} \tag{40}$$

Based on the user's choice of the number of particles and η, l_0 is found. The points along the segmented throat outline are then redistributed in such a way that the mean spacing matches l_0. Using this new outline, three further extruded areas are created, using the same spacing, in order to create a similar region padding as the one shown in 21. The domain is then sequentially filled with particles. This process is shown in figure 25.

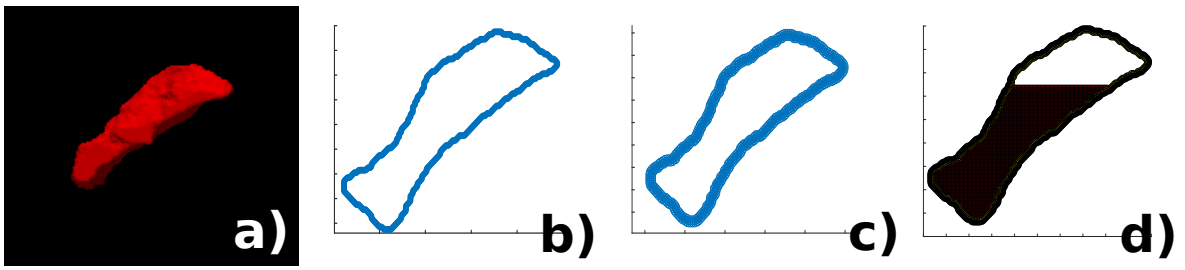


Figure 25: Creating a uniform particle layout over a complex boundary. a) 3D segmented mouth geometry, b) 2D slice, c) Using PolyArea in Matlab to create outline rediscretised layers, and d) Importing the geometry to the solver

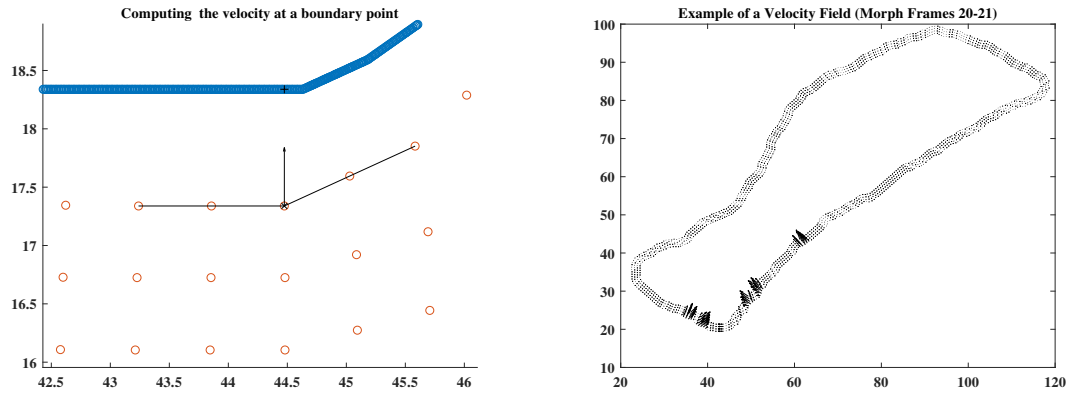
5.4 Moving the Boundary

Having illustrated the capacity of the solver to cope with various benchmark cases, moving boundary simulations are attempted. By importing the morphed surface CSV files into MatLab, an expected boundary is generated at each morphing time step. The time step between two medical frames is $dt_{frame} = 0.175sec$ and for a number of morphing steps, N_{steps}, $dt_{morph} = \frac{dt_{frame}}{N_{steps}-1}$. The velocity for each of the boundary particles is computed as:

$$u_i^n = \frac{min(\bar{x}_j^{n+1} - x_i)}{dt_{morph}} \tag{41}$$

where n indicates a morph time instance, i is a boundary point and j is a point along the new boundary. Each layer of the new boundary surface, x_j^{n+1}, was rediscretised temporarily to have 10 times the number of points of the previous time instance, labeled $\bar{x}$. With this the distance in the changing surface direction is computed. The method is illustrated in figure 26a, with the $min()$ signifying the closest point to the temporary surface. Figure 26 shows a computed velocity field. Although this method works for simpler cases, when used to represent convex translations such as the ones in swallowing, the wall particles distort leading to an unnaturally high repulsive pressure. To address this, the wall particles could be substituted by a Z function. Z is a distance based look-up table, where the n density values for a particle at different distances from the wall are pre-calculated. Thus, the particle density near a boundary can be re-written as:

$$n_i = n_f + n_w \text{ and } n_w = Z(r_{wall}) \tag{42}$$



(a) (b)

Figure 26: a) Illustration of the velocity computation and b) resulting velocity field

where, r_{wall} is the closest distance to a boundary. This method has been shown to yield very good results in recent literature [33, 40, 41]. The implementation of such a method together with the current fluid solver basis is part of a future investigation.

6 PREPROCESSING AND CFD CALCULATIONS

STARCCM+ is used to generate a volume mesh based on an initial stationary surface mesh; typically, this is used to perform CFD calculations. However, to implement a moving boundary simulation, it is necessary to morph the geometry, and hence the mesh, with respect to time. This section incorporates the CSV file from the morphing process to perturb the mesh and obtain meaningful CFD results.

Macro Generation and Import into STARCCM+

When importing the surface mesh into STARCCM+, a 3D volume is needed. An STL file is generated for the initial stage of the simulation ($t = 0$), and a volume mesh is generated. A base size of 2m is used, as this provided a balance between geometric accuracy and computing time. This value will be analysed later using the final simulation.

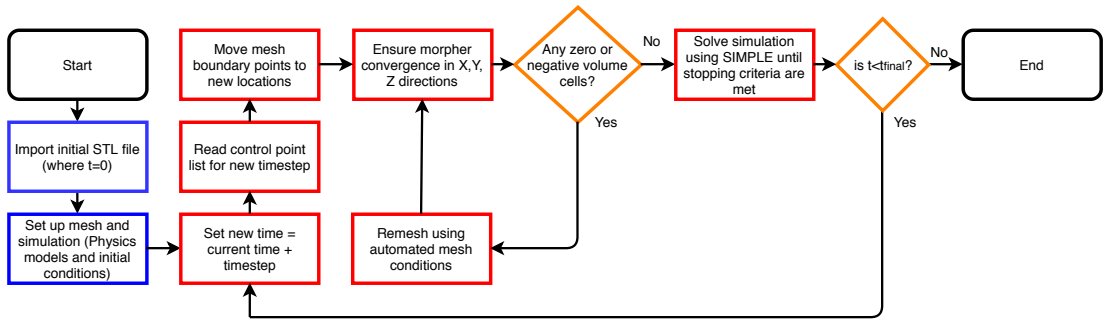


Figure 27: Process of the Java Macro. Blue section is used for initialisation. The red section includes the looped process

Once the initial volumetric mesh is created, the CSV files are used to morph the mesh; these outline the position of the bolus boundary points for a particular time. The method is to produce a mesh from which the CSV files can be compared to at each iteration. A macro is used to automate this process, to alleviate human input, because there are 2141 timesteps. They are particularly useful for repetitive operations and improve the efficiency of the process [42].

The implementation of a macro is relatively straightforward since they are a function built into STARCCM+. The process is shown in figure 18.

In order to run the macro through the STARCCM+ client, it is necessary to include the Java APIs (The Javascript used to run STARCCM+) in the macro script. Within STARCCM+, the morpher specification for the region was set to floating because this reduces the formatting needed for the data provided. This process will however have to be repeated for each time step, but since it is a repetitive augmentation a loop can be used.

The initial conditions for the bolus are set using volume fractions of 0.7 and 0.3 for honey and air respectively. The physics conditions are chosen as a laminar, multiphase flow. With the time initialised, the Java file reads the CSV file appropriate for the current time value and update the control points. If the new mesh is acceptable the macro progresses; at this point the simulation is solved. The solvers are outlined in the following section. The time value is then increased by 1ms and continues until the final time.

7 FINITE VOLUME METHODS

7.1 STARCCM+ Solvers

STARCCM+ uses the Finite Volume method in order to discretise the region. This allows the computation of the conservation equations by splitting these equations into space and time [43]. The generated mesh is used for the spatial discretisation, while the specified time step is used for the time discretisation. The time step is determined by the total time divided by the number of frames for each stage of the simulation. Since there are 22 images from the segmentation, and 100 interpolations between each image which produces a time step of 1ms. This is a small timestep which aids in the accuracy and stability of the calculations as the implicit, unsteady solver is used. Since the geometry varies considerably, and has a large effect on the fluid, this simulation is likely to be highly unstable and hence the implicit model is being used. In order to compute the pressure and velocity fields within the region, a solver needs to be chosen. The implicit, unsteady conditions combined with the segregated flow solver, allows the use of the SIMPLE algorithm to iteratively calculate the gradient fields within each control volume [44]. This allows the calculation of the conservation equations (mass, energy, and momentum) across each boundary between every control volume, or cell [45] [44]. The conservation equations can be obtained from the general transport equation:

$$\frac{d}{dt}\int_V \rho\phi dV + \int_A \rho\mathbf{v}\phi.d\mathbf{a} = \int_A \Gamma\nabla\phi d\mathbf{a} + \int_V S_\phi dV \tag{43}$$

Which includes: the Transient term, Convective flux, Diffusive flux, and the Source term respectively [46]. Moreover, ϕ represents a scalar property, unique to each conservation equation, A is the surface area of the control volume, and d$\mathbf{a}$ represents the surface vector. The SIMPLE algorithm is shown in figure 28.

Where ω is the relaxation factor which is used to reduce the probability of the pressure correction factor, p', being overestimated. The discretised momentum and pressure correction equations used are as follows:

$$\frac{\mathbf{u}_0 - \mathbf{u}^n}{\Delta t} = -\mathbf{u}_0\nabla\mathbf{u}_0 + \alpha\nabla^2\mathbf{u}_0 - \frac{\nabla p_0}{\rho} \tag{44}$$

$$\nabla^2 p' = \frac{\rho}{\Delta t}(\nabla\mathbf{u}_0) \tag{45}$$

To improve convergence and stability, there will be 10 inner iterations for each step in order for the pressure and velocity corrections to minimise the errors. An additional threshold criterion was chosen to ensure the momentum residuals, in the X,Y, and Z directions, reduce below a threshold value of $1x10^{-6}$ prior to moving to the next timestep. This should allow sufficient convergence, however, this will be analysed in the final simulation. By using a length scale of 2 from the mesh, and the material properites outlined previously, a Reynolds number of around 1 is produced. Since the Reynolds number is low, it is possible to assume that the flow is laminar and so no turbulence modelling is required. This low Reynolds number, and velocity, allows the assumption of constant density, or incompressible flow [47].

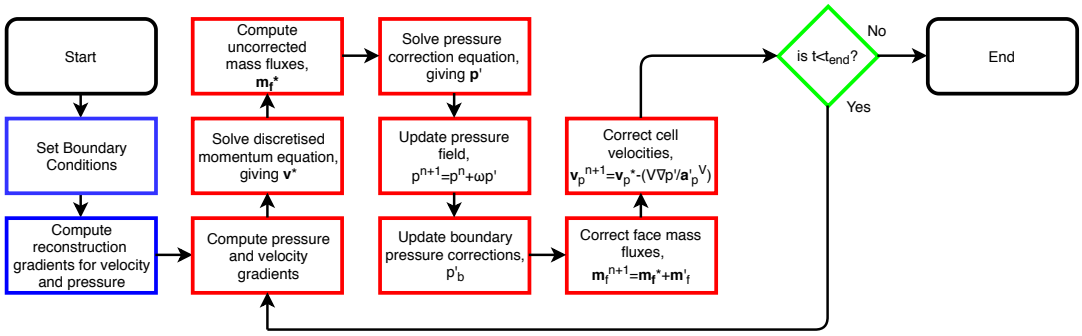


Figure 28: Process of the SIMPLE algorithm. Blue section is used for initialisation. The red section includes the looped process

7.2 Validation - Dam Break

It was important to validate the use of the proposed models for the final simulations. The first of two simulations chosen is a dam-breaking case to illustrate the use of the the multiphase flow solver, Volume of Fluid (VoF) [48]. This uses an Eulerian reference frame rather than a Langrangian one. This model concentrates on the interface between the two phases and is particularly useful for free surface flows. The model uses a scalar quantity from 0 to 1 to quantify the amount of fluid within each cell. As a gravity driven flow, this will help to validate the model in the main simulation. The simulation was run for 10 seconds, with timesteps of 5ms. The scalar images of the volume fraction of water are shown in figure 19.

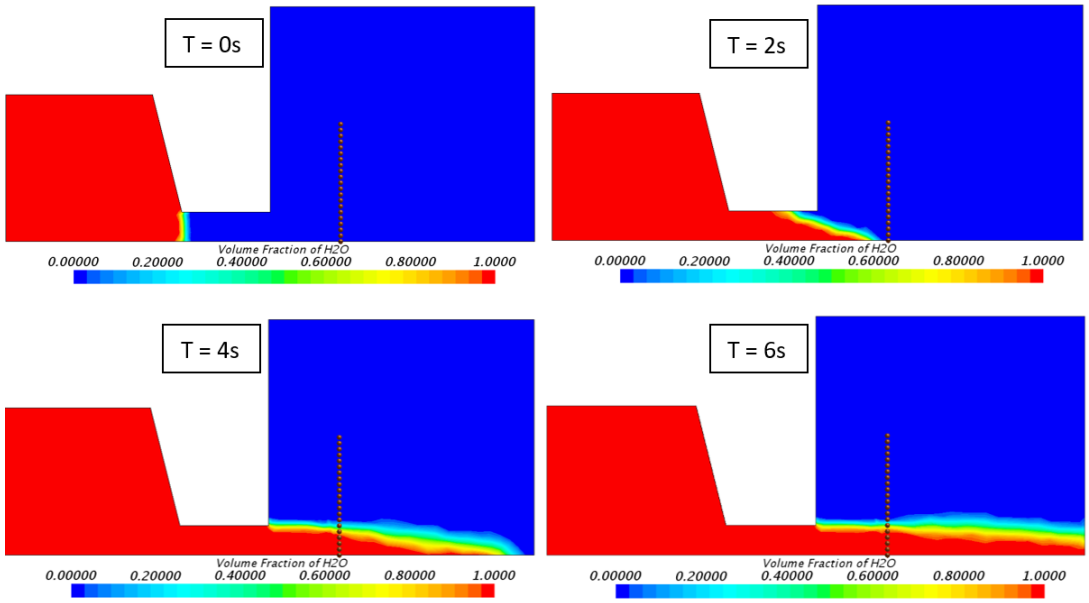


Figure 29: Results of Dam Breaking simulation

V. V. Degtyarev et al.[49] produced a direct comparison between experimental and analytical solutions for a dam-break system involving a sudden change in cross-sectional area. This was used to validate the results from the dam-break problem.

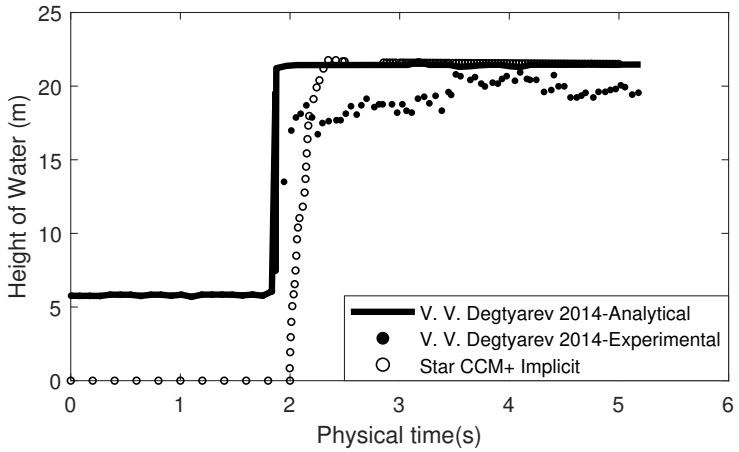


Figure 30: Comparison between literature (Degtyarev et al. 2014) and testing simulation

Figure 30 shows the comparison between the fluid level of the water 1.7m from the initial imaginary dam. It can be seen from the figure that there is a strong correlation between the research and the validation simulation and a similar settling time. Therefore, this method can be deemed an accurate model for free surface and gravity driven flows.

7.3 Validation - Contracting Cylinder

The second simulation tests internal flows through a morphing cylinder. This was undertaken to demonstrate the use of field functions to produce a morphing mesh. The simulation was a laminar flow of liquid (water) through the cylinder, which itself had an induced velocity. This velocity was set to -0.25 ms^{-1} in the middle of the cylinder in the radial direction, and decreased to zero as it approached the ends of the cylinder. The inlet velocity of the fluid was set to 10 ms^{-1}.

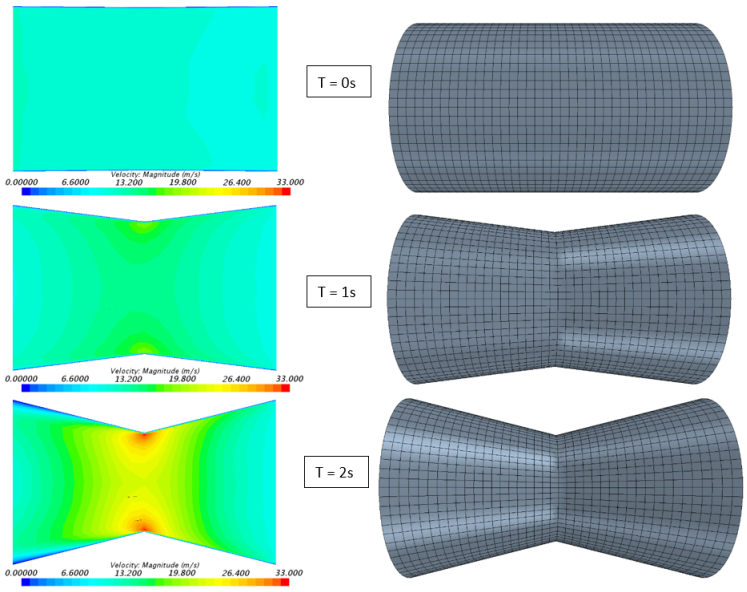


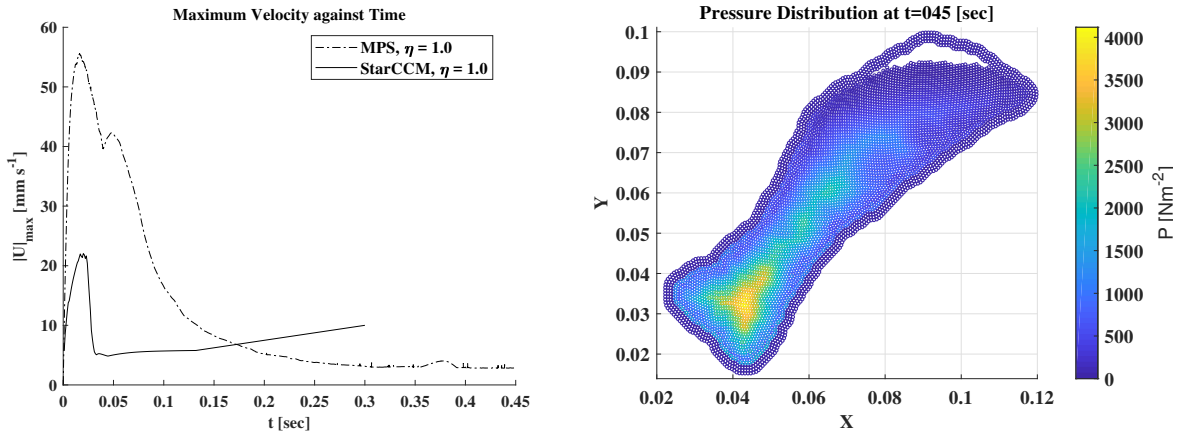
Figure 31: Results of Cylinder Contraction simulation

The resulting velocity profile of the fluid along the centre of the cylinder is shown alongside the cylinder mesh in figure 31. It can be seen that the deformation of the wall has a significant impact on the velocity of the liquid, especially in the centre. However, this simulation continued to run while maintaining momentum residuals below $1x10^{-4}$, and therefore is a viable model to use.

8 FLUID SIMULATION RESULTS

8.1 Comparison of STARCCM+ and MPS for a static simulation

The results of the two fluid solvers are compared in a static simulation of frame 6. The mouth cavity is simulated as being full ($\eta = 1$) and the resulting velocity and pressure fields are compared. The substance used for the static tests and for the STARCCM+ moving simulations is honey to replicate what the patient was swallowing[50].



(a) (b)

Figure 32: a) Maximum velocity against time for the solvers, b) MPS pressure field.

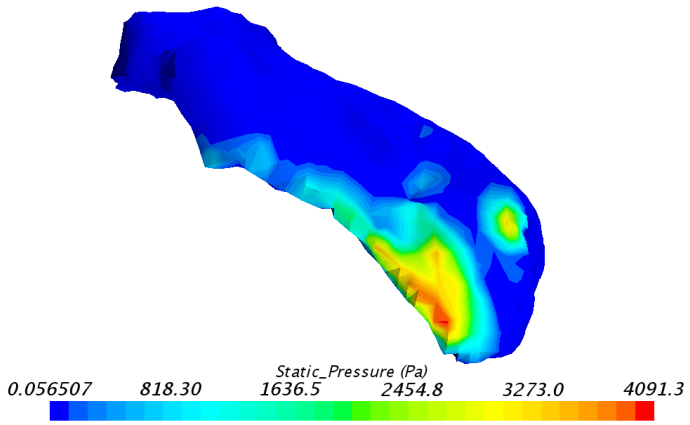


Figure 33: STARCCM+ static pressure field

Figure 32a shows the stationary performance of the two solvers, the MPS pressure field and the STARCCM+ pressure.

The pressure field for both STARCCM+ and the MPS method are very similar in terms of both the location of the high pressure regions and the magnitude of the pressure (figure 32b and 33). Despite these similarities, it is found that the STARCCM+ solver is prone to instabilities. This phenomenon is illustrated in figure 32a, where until $t = 0.05s$ the solver

residuals are decaying and the velocity decreases. However, after that point the numerical simulation becomes unstable, as both the velocity increases and the solver residuals oscillate.

8.2 Swallowing Simulation STARCCM+

With a combination of the Volume of Fluid and moving boundary models and techniques, as well as the macro functions, the simulation is run for volume fractions of 0.7 and 0.3, for honey and air respectively. The following results show the mesh of the bolus and the volume fraction of the liquid within the bolus at each 0.2s timestep.

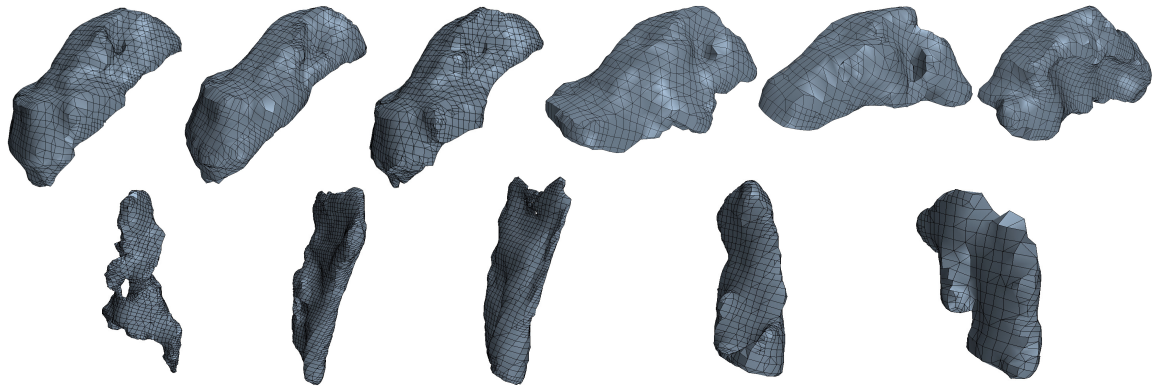


Figure 34: Mesh of the Bolus at 0.2s intervals

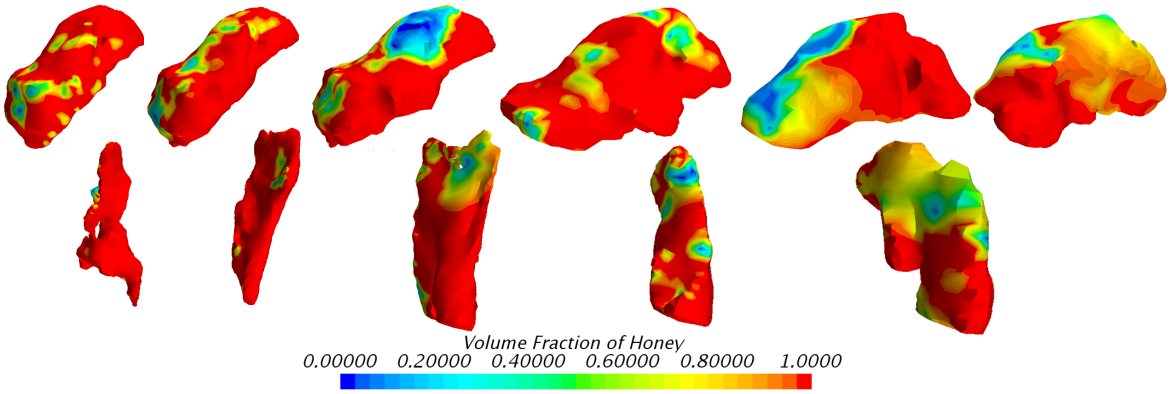


Figure 35: Behaviour of liquid during the swallowing process

As can be seen from the images of the mesh, the geometry of the bolus varies considerably during the swallowing process. These show the process occurring during the swallow, allowing for easy detection of medical problems. Comparing the generated meshes to the segmentations shows that the mesh at each timestep is a sensible approximation to the bolus. The moving mesh, created using the macro, morphed without issue. However, there are instances where the mesh as a whole moves within the reference frame which causes some complications with the velocities within the bolus. These large displacements cause long computation times however, it does not result in any errors. This reiterates the validity of this method.

The volume fraction of honey images show how the liquid behaves as the bolus is morphing and progressing through the throat. In the first few timesteps, where the movement is minimal, the air settles to the top of the mouth as would be expected with a static case. However, as the process continues, there is more mixing between the air and honey, resulting in less defined boundaries between the fluids. This can be seen especially in the final few frames where the

initial large air bubble has been mixed into the liquid, creating smaller air pockets throughout the bolus.

With the addition of the liquid, there were some stability issues created by the large mesh deformations along with the large jumps in the mesh location within the domain. With the timestep of 1ms used, the majority of the simulations run without significant change in the residuals. However, in cases where the bolus would alter significantly, this cases a large increase in the velocities and pressures and hence, the continuity residuals of the liquid. This results in an uncontrollable increase in the residuals and so there is instability. This could be controlled by ensuring the centre of the bolus is stabilised throughout the simulation; this would guarantee all that induced velocities on the fluid are purely due to the mesh deformation and not the movement during segmentation. If a significant amount of time could be devoted to these simulations, another improvement that could be made would be to have a much smaller timestep. This would require more interpolations between each medical image, and running the simulation using this smaller value. This would aid in reducing the impact of the large deformations and hence, the stability of the simulation, allowing for a more robust system.

As a model for comparison, this is a suitable solution, and would aid in diagnosing specific issues or pinpointing the variation between a healthy patient and a patient with swallowing difficulties. There are however, improvements that can be made to improve the stability of this model.

9 CONCLUSION

The aim of this project was to develop the basis of an automatic CT scan based swallowing simulation method. To achieve this, the bolus of a healthy patient's swallow is successfully segmented. Subsequently, by using a bi-cubic time interpolation, these results are morphed in order to create a smoothly moving bolus and throat geometry compatible with the CFD solvers. With respect to modelling the bolus movement, a novel Lagrangian MPS simulator has been developed and a STARCCM+ Volume of Fluid based method. The MPS solver is successful in passing a series of benchmark tests. However, stability issues arise in the moving boundary method; a possible solution to this is proposed. The STARCCM+ solver performs well on the moving mesh benchmark of the contracting cylinder and the bolus mesh movement. Alas, with the inclusion of the liquid the solver fails to consistently produce stable and reliable results. Some useful numerical data, like the static pressure fields, were obtained which can provide key information when diagnosing Oropharyngeal Dysphagia. Although further research is needed to achieve results for dynamic velocity fields, this report provides a platform for such work.

REFERENCES

[1] E. Del Giudice, A. Staiano, G. Capano, A. Romano, L. Florimonte, E. Miele, C. Ciarla, A. Campanozzi, and A. F. Crisanti, "Gastrointestinal manifestations in children with cerebral palsy," *Brain and Development*, vol. 21, no. 5, pp. 307–311, 1999.

[2] P. C. Langdon, A. H. Lee, and C. W. Binns, "Dysphagia in acute ischaemic stroke: severity, recovery and relationship to stroke subtype," *Journal of Clinical Neuroscience*, vol. 14, no. 7, pp. 630–634, 2007.

[3] M. D. Good-Fratturelli, R. F. Curlee, and J. L. Holle, "Prevalence and nature of dysphagia in va patients with copd referred for videofluoroscopic swallow examination," *Journal of communication disorders*, vol. 33, no. 2, pp. 93–110, 2000.

[4] R. W. Walker, J. R. Dunn, and W. K. Gray, "Self-reported dysphagia and its correlates within a prevalent population of people with parkinson's disease," *Dysphagia*, vol. 26, no. 1, pp. 92–96, 2011.

[5] M. Airoldi, M. Garzaro, L. Raimondo, G. Pecorari, C. Giordano, A. Varetto, P. Caldera, and

R. Torta, "Functional and psychological evaluation after flap reconstruction plus radiotherapy in oral cancer," *Head & neck*, vol. 33, no. 4, pp. 458–468, 2011.

[6] K. J. Aldridge and N. F. Taylor, "Dysphagia is a common and serious problem for adults with mental illness: a systematic review," *Dysphagia*, vol. 27, no. 1, pp. 124–137, 2012.

[7] S. B. Leder, B. L. Judson, E. Sliwinski, and L. Madson, "Promoting safe swallowing when puree is swallowed without aspiration but thin liquid is aspirated: Nectar is enough," *Dysphagia*, vol. 28, no. 1, pp. 58–62, 2013.

[8] C. S. Lear, J. Flanagan Jr, and C. Moorrees, "The frequency of deglutition in man," *Archives of Oral Biology*, vol. 10, no. 1, pp. 83–IN15, 1965.

[9] W. J. Dodds, E. T. Stewart, and J. A. Logemann, "Physiology and radiology of the normal oral and pharyngeal phases of swallowing." *AJR. American journal of roentgenology*, vol. 154, no. 5, pp. 953–963, 1990.

[10] R. Martino, N. Foley, S. Bhogal, N. Diamant, M. Speechley, and R. Teasell, "Dysphagia after stroke: incidence, diagnosis, and pulmonary complications," *stroke*, vol. 36, no. 12, pp. 2756–2763, 2005.

[11] W. Edgar, "Saliva: its secretion, composition and functions," *British dental journal*, vol. 172, no. 8, p. 305, 1992.

[12] N. Diamant, "Physiology of esophageal motor function." *Gastroenterology clinics of North America*, vol. 18, no. 2, pp. 179–194, 1989.

[13] J. B. Palmer, N. J. Rudin, G. Lara, and A. W. Crompton, "Coordination of mastication and swallowing," *Dysphagia*, vol. 7, no. 4, pp. 187–200, 1992.

[14] J. A. Logemann, "Evaluation and treatment of swallowing disorders," *American Journal of Speech-Language Pathology*, vol. 3, no. 3, pp. 41–44, 1994.

[15] G. Ergun, P. Kahrilas, S. Lin *et al.*, "Pattern and modulation of tongue surface movement during deglutition," in *Dysphagia Research Society (DRS) Annual Meeting. Milwaukee, WI*, 1992.

[16] B. Martin-Harris, M. B. Brodsky, Y. Michel, F.-S. Lee, and B. Walters, "Delayed initiation of the pharyngeal swallow: normal variability in adult swallows," *Journal of Speech, Language, and Hearing Research*, 2007.

[17] R. Shaker, W. J. Dodds, R. O. Dantas, W. J. Hogan, and R. C. Arndorfer, "Coordination of deglutitive glottic closure with oropharyngeal swallowing," *Gastroenterology*, vol. 98, no. 6, pp. 1478–1484, 1990.

[18] F. M. Mcconnel, "Analysis of pressure generation and bolus transit during pharyngeal swallowing," *The Laryngoscope*, vol. 98, no. 1, pp. 71–78, 1988.

[19] F. J. Ingelfinger, "Esophageal motility," *Physiological reviews*, vol. 38, no. 4, pp. 533–584, 1958.

[20] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[21] A. M. Gambaruto, "Processing the image gradient field using a topographic primal sketch approach," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 31, 2015.

[22] I. Selesnick, "Four Types of Linear-Phase FIR Filters," 2005.

[23] C. Hirsch, *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*, 2007.

[24] P. Holoborodko, "Smooth Noise Robust Differentiators," 2008. [Online]. Available: http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/

[25] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[26] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[27] D. Ashlock, "Morphing shapes, morphing rules," Feb 2019. [Online]. Available: https://occupymath.wordpress.com/2019/02/07/morphing-shapes-morphing-rules/

[28] E. F. Krause, *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation, 1986.

[29] A. Koshakji, "Free form deformation techniques for 3d shape optimization problems," 2011.

[30] Y. Saito, H. Kato, M. Otsuki, I. Kimura, Y. Shimizu, and E. Isenko, "Refinement of MPS method for practical application to snow avalanches," *Annals of Glaciology*, vol. 61, no. 53(61), 13-22, pp. 13–22, 2012.

[31] S. Koshizuka and Y. Oka, "Moving-Particle Semi-Implicit method for fragmentation of incompressible fluid," *Nuclear Science and Engineering*, vol. 123, pp. 421–434, 1996.

[32] A. M. Gambaruto, "Computational haemodynamics of small vessels using the Moving Particle Semi-implicit (MPS) method," *Journal of Computational Physics*, vol. 302, pp. 68–96, 2015.

[33] S. Koshizuka, K. Shibata, M. Kondo, and T. Matsunaga, *Moving Particle Semi-Implicit Method, A Meshfree Particle Method for Fluid Dynamics*, 2018.

[34] M. Oochi, S. Koshizuka, and M. Sakai, "Explicit MPS Algorithm for Free Surface Flow Analysis," *Transactions of the Japan Society for Computational Engineering and Science*, vol. 2010, 2010.

[35] A. Fadafan and H. Kermani, "Moving particle semi-implicit method with improved pressures stability properties," *Journal of Hydroinformatics*, vol. 6, no. 20, pp. 1268–1285, 2018.

[36] H. Ding, C. Shu, K. Yeo, and D. Xu, "Development of least-square-based two-dimensional finite-difference schemes and their application to simulate natural convection in a cavity," *Computers Fluids*, vol. 33, no. 1, pp. 137–154, 2004.

[37] A. Khayyer and H. Gotoh, "Modified Moving Particle Semi-implicit methods for the prediction of 2D wave impact pressure," *Coastal Engineering*, vol. 56, no. 4, pp. 419 – 440, 2009.

[38] Y. Yamada, M. Sakai, S. Mizutani, S. Koshizuka, M. Oochi, and K. Murozono, "Numerical Simulation of Three-Dimensional Free-Surface Flows with Explicit Moving Particle Simulation Method," *Atomic Energy of Japan Internal Publications*, vol. 10, no. 3, pp. 185–193, 2010.

[39] P. K. Stansby, A. Chegini, and T. C. D. Barnes, "The initial stages of dam-break flow," *Journal of Fluid Mechanics*, vol. 374, pp. 407–424, 1998.

[40] H. Takahiro, K. Seiichi, and K. Yoichiro, "Smoothed Particle Hydrodynamics in Complex Shapes," pp. 191–197, 2007.

[41] T. HARADA, S. KOSHIZUKA, and K. SHIMAZAKI, "Improvement of Wall Boundary Calculation Model for MPS Method," *Transactions of the Japan Society for Computational Engineering and Science*, vol. 2008, pp. 20 080 006–20 080 006, 2008.

[42] Techopedia, "Macro Definition," 2010. [Online]. Available: https://www.techopedia.com/definition/3833/macro

[43] M. Peric, R. Kessler, and G. Scheuerer, "Comparison of finite-volume numerical methods with staggered and colocated grids," *Computers and Fluids*, vol. 16, no. 4, pp. 389 – 403, 1988.

[44] J. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, 2002.

[45] I. Demirdzic and S. Muzaferija, "Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology," *Computer methods in applied mechanics and engineering*, vol. 125, pp. 235 – 255, 1995.

[46] S. W. Bird, R.B. and E. Lightfoot, *Transport Phenomena*, 1966.

[47] J. Y. M. S. R. Mathur, "Pressure boundary condition for incompressible flow using unstructured meshes," *Numerical Heat Transfer*, vol. 32, no. 3, pp. 283 – 298, 1997.

[48] S. Muzaferija and M. Peric, "Computation of free-surface flows using the finite-volume method and moving grids," *Numerical Heat Transfer*, vol. 32, no. 4, pp. 369 – 384, 1997.

[49] V. V. Degtyarev, V. V. Ostapenko, O. A. Kovyrkina, and A. V. Zolotykh, "Comparison of theory and experiment in simulation of dam break in a rectangular channel with a sudden change in cross-sectional area," *Journal of Applied Mechanics and Technical Physics*, vol. 55, no. 6, pp. 999–1004, 2014.

[50] ANTON-PAAR LtD, "Viscosity of Flower Honey (Blended), using the SVM viscosity measurement apparatus," 2019.